TIME-OF-ARRIVAL ESTIMATION FOR SATURATED OPTICAL TRANSIENTS USING CENSORED PROBABILISTIC MODELS

A Thesis
Presented to
The Academic Faculty

By

Matthew J. Kagie

In Partial Fulfillment
of the Requirements for the Degree
Masters of Science in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

December 2016

TIME-OF-ARRIVAL ESTIMATION FOR SATURATED OPTICAL TRANSIENTS USING CENSORED PROBABILISTIC MODELS

Approved by:

Dr. Lanterman, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Romberg
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Citrin
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: December 7, 2016

This thesis is dedicated to my grandfather, Gus Triantos.

ACKNOWLEDGEMENTS

I would first like to thank my thesis advisor, Professor Lanterman of the School of Electrical and Computer Engineering of Georgia Institute of Technology. Throughout my time at Georgia Tech, he has been a great source of technical knowledge and literary refinement.

I would also like to acknowledge Professors Romberg and Citrin, also of the School of Electrical and Computer Engineering at Georgia Institute of Technology, as the other readers on my thesis committee. I am very grateful for their valuable comments on this thesis.

I would like to thank Mike Skaggs, my manager at Sandia National Laboratories, as well as my mentor, Park Hays, also of Sandia, for their support and guidance in my research.

Finally, I would like to express my profound gratitude to my parents and my brother, whose support and encouragement throughout the process of writing this thesis has been absolutely unwavering. This accomplishment would not have been possible without them. Thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The objective of the proposed research is to estimate the time-of-arrival of a transient optical signal subjected to a particular type of nonlinear distortion. The limited dynamic range of optical sensors can result in nonlinear distortion when measuring extreme transient events, such as lightning. To deal with saturated signals, we employ censored probabilistic models to develop maximum-likelihood procedures for estimating the time-of-arrival of lightning strikes, along with associated nuisance parameters. The received signal is modeled as a realization of a Poisson point process characterized by parametric models of a lightning strike's time-varying intensity. The models are extracted from the FORTÉ lighting database via machine learning techniques. Using Monte Carlo simulations, we compare the variances of different algorithms as a function of signal magnitude and saturation threshold. We also compare these variances to analytical performance bounds such as the Cramér-Rao lower bound.

CHAPTER 1
INTRODUCTION AND BACKGROUND

1.1 History of the Problem

Parameter estimation plays an important role in many areas of electrical engineering, including digital communications, radar, and sonar [1]. One key parameter is the time-of-arrival (TOA) of a received signal. The received signal is usually modeled as a realization of a random process with a parameterized probabilistic model. While there are many possible methods for estimating these parameters, the maximum-likelihood (ML) estimator, which seeks the parameters that maximize the probability density for a particular set of data [2], has many appealing properties and is conceptually elegant.

A traditional use of ML estimation can be seen in [3], which models the received signal as a combination of a well-known signal (with some unknown time shift) and a noise component (modeled as additive white Gaussian noise (AWGN)), hoping that this noise component will accommodate all interference and multipath characteristics without needing to explicitly estimate them.

The authors of [4] adapt this approach to estimating the TOA of an optical pulse received by a photon-counting photodetector. Their model includes the binning of photon counts at the receiver. They employ ML with a Poisson model for estimating the TOA. This model of a receiver as a photon counter is also prevalent in LIDAR/LADAR [5, 6]. We have adopted the model of a photon-counting receiver in our research.

The authors of [7] operate on signals that are modeled as filtered Poisson processes, for which ML estimators are difficult to compute. They employ an iterative EM algorithm, in which the E-step calculates the optimal linear estimate of the point process as the output of a linear-time varying filter. Our research also employs EM algorithms.

A more recent TOA approach is to use machine learning algorithms. The authors of [8] move entirely away from the idea of physics-inspired ML techniques. They address the issue of detection and TOA estimation of underwater acoustic signals, where the signal length and structure are unknown a priori, and there are so many perturbations and distortions that it is difficult to create a reasonable model to use with ML. They use clustering techniques to determine if incoming data is signal or noise, and they use the first sample of any waveform labeled "signal" as the TOA.

ML usually works well when an accurate model of the received signal is known, but may fail when the signal undergoes some unmodeled distortion. The filtered Poisson process approach of [7] attempts to correct for such distortions by modeling the system with a linear time-varying filter. In this thesis, we address the estimation of the TOA given the nonlinear distortion of saturation. This form of signal corruption is not well suited for the techniques described in [7]. The machine learning approach of [8] is interesting in that it throws away preconceived models of the incoming signal, but we feel that physically informed, model-based ML approaches are more appropriate for addressing saturation.

1.2 Overview of the Research

We develop TOA estimation algorithms for optical transient signals (OTSs) that have undergone the nonlinear distortion of saturation. We first use machine learning to analyze a database of lighting strikes to generate lightning templates. These lightning templates act as the OTSs whose TOAs we attempt to estimate. Next, we develop ML algorithms based on several modifications of Poisson and Gaussian probability models, in which the mean and variances of Gaussian distributions are chosen to match the mean of Poisson distributions. These manipulations include the original distributions, censored distributions, and an "ad hoc" method. We also explored Gaussian weighted least squares (WLS) approximations, which typically result in calculations that are greatly simplified as a result of replacing variance parameters in the Gaussian form with the square of the measured data. However, upon

pursuing this method, it became apparent that the addition of the saturation negates the typical simplifying effects of the WLS approximation, and actually causes this methodology to become more complex than the Poisson and Gaussian cases without the WLS approximation. Hence, the WLS approach has been left out of the thesis. Our ML algorithms estimate the TOA, as well as the nuisance parameter of scaling factor. We also find theoretic lower bounds for the variances of each of these estimators (in the form of the Cramér-Rao lower bound). Finally, we run Monte Carlo simulations and discuss the overall effectiveness of each of these methods.

1.2.1 Lightning Model

Some satellites orbiting Earth's surface are equipped with optical transient sensors, providing information about phenomena such as the frequency and spatial distribution of lightning strikes [9]. The OTSs explored in this thesis are lightning strikes. Most existing models for lightning [10, 11] are either too simple or overly complicated and highly parameterized (implying a large numbers of nuisance parameters). Hence, we decided to construct our own models using machine learning techniques.

The FORTÉ satellite collects optical and radio-frequency transient signals from lightning. A photodiode detector (PDD) detects and records optical data [12]. This database has been used for other purposes, such as modeling photon transport through clouds [13].

We use one year's worth of FORTÉ PDD data, from 2001. Many waveforms in this database do not correspond to actual lightning strikes, but instead contain various kinds of clutter. Hence, we first need to separate the lightning signals from non-lightning signals to be able to cull the non-lightning signals from the database.

As described in Section 2.3, we use an unsupervised clustering algorithm to aggregate the waveforms into groups. Then, we marked several of the groups as non-lightning by manually inspecting a random sampling of waveforms from each group. Marking groups as either lightning or non-lightning is a supervised binary classification operation. We then

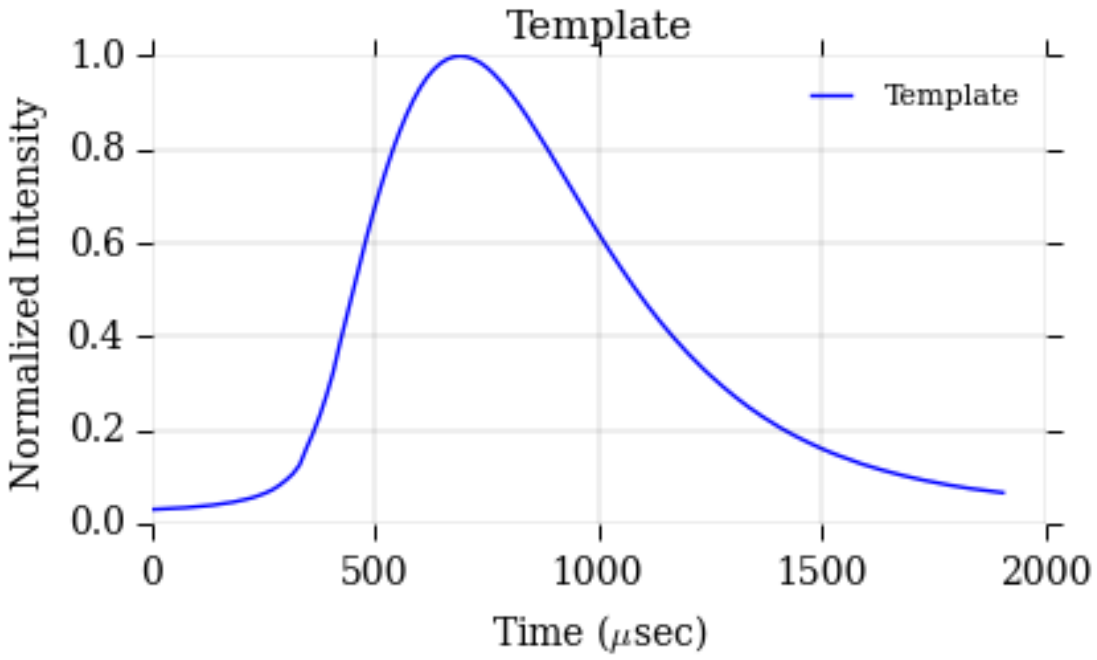

Figure 1.1: Example of a lightning template created using unsupervised machine learning on a large database of lightning strikes.

applied this "trained" classifier to the entire database.

After removing the signals from the non-lightning groups from further consideration, we clustered the remaining signals into families by leveraging nonnegative matrix factorization. This resulted in nine distinct families of lightning signals, as well as a family of likely "non-lightning" signals that had passed the first culling stage.

Finally, we created lightning "templates" that consisted of the average signal of each of the aforementioned families. The template shown in Fig. 1.1 is used in the MC simulations described later.

1.2.2 Received Signal Model

Our work explores lightning strikes received by an optical transient sensor acting as a photon counter, as illustrated in [14]. There are a finite number of photons collected at each sample in time, and the expected number of photons is ideally proportional to the intensity

of the received signal at that time. Hence, the received signal is modeled as a nonhomogeneous Poisson point process (PPP). PPP are used to model random realizations of points in time, space, or a combination of the two. Our research focuses on random realizations of points in time [15].

We model the signal as a realization of a PPP with the time-varying intensity:

$$\lambda(t) = \alpha\mu(t - \tau) + \lambda_0, \tag{1.1}$$

where α is a nonrandom scale parameter that must be estimated (generally a nuisance parameter in our application), μ is the aforementioned template, time-shifted by the TOA τ, and λ_0 is the background photon level[1] of the optical sensor, which is also assumed known (perhaps though calibration measurements). Since we are working with discrete-time signals, (1.1) is discretized as

$$\lambda_i = \alpha\mu_i^{(\tau)} + \lambda_0, \tag{1.2}$$

where $\mu_i^{(\tau)}$ is the i$^{\text{th}}$ time sample of the template time-shifted by τ.

Due to the large dynamic range of OTSs, it is difficult to design a sensor to accurately measure values across the entire practical range of intensities. This introduces nonlinear distortions such as saturation, where OTSs with values greater than a sensor-specific threshold are clipped and recorded incorrectly, which complicates parameter estimation. This thesis will use the terms "saturated" and "censored" interchangeably.

We will denote random variables using upper case letters and realizations of those random variables with equivalent lower case letters. Let $\mathbb{R}^+$ represent the set of nonnegative real numbers. We model the effect of saturation as follows: for an ideal unsaturated discrete-time signal $y \in (\mathbb{R}^+)^N$, the received signal (after being subjected to the sensor's

[1]While we treat λ_0 as a constant, the mathematics in this thesis could be easily extended to it being a known time-varying function.

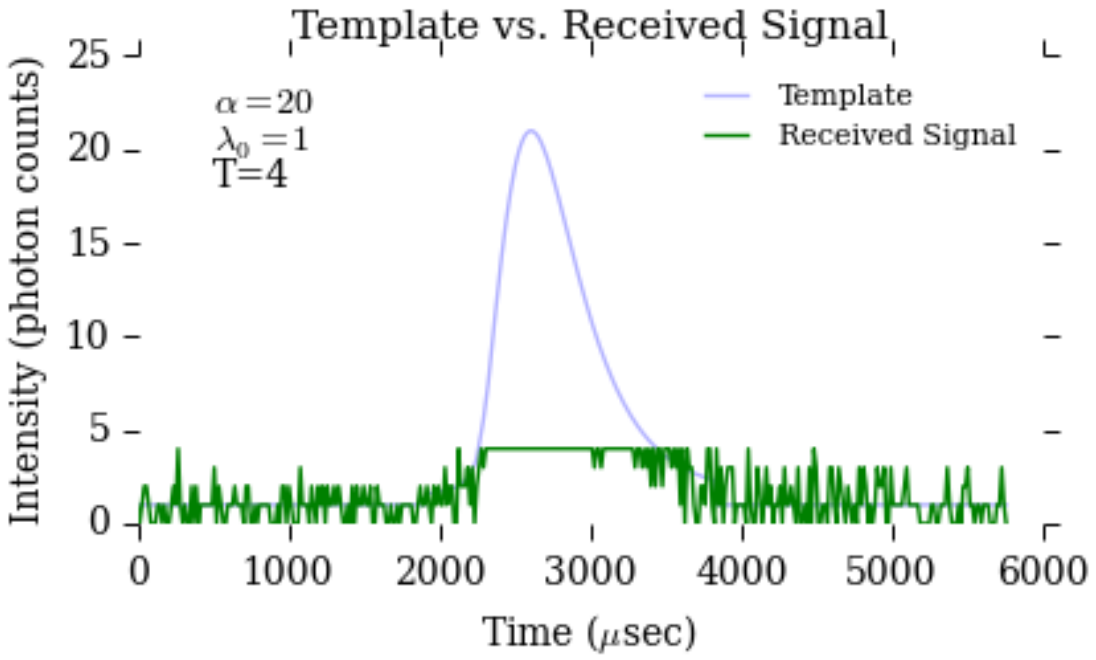

Figure 1.2: Example of a synthesized received signal. The received signal is a random realization of a nonhomogeneous Poisson point process with time-varying intensity λ that has been saturated at the sensor-specific saturation threshold, T.

saturation) $x \in (\mathbb{R}^+)^N$ is

$$x_i = \min(y_i, T), \;\; 1 \leq i \leq N, \tag{1.3}$$

where T is a known sensor-specific saturation threshold, x_i is the i$^{\text{th}}$ time sample of the signal x, and y_i is the i$^{\text{th}}$ time sample of the signal y. An example of a template, λ, as well as a received signal undergoing saturation can be seen in Fig. 1.2.

1.2.3 Estimators

This section discusses traditional probabilistic models for optical transient signals used for detection and parameter estimation, particularly Poisson and Gaussian likelihoods. For each underlying probability model, in addition to properly censored distributions, an "ad hoc" method is formulated and explored. In the ad hoc methods, the algorithms appropriate

for the uncensored models are used, but the modeled intensity, i.e. λ, is artificially saturated at the known saturation threshold.

Poisson

It is natural to model a recorded optical data value as a realization of a Poisson random variable, with the different values comprising a time-domain signal being independent of each other when conditioned on underlying parameters. Therefore, the probability mass function (PMF) for a given signal is the product of the PMFs for the individual values for different time samples.

In the absence of a reasonable prior distribution for the parameters, maximizing the loglikelihood is a reasonable approach. To solve for the TOA using the Poisson distribution, this optimization program becomes

$$\hat{\tau} = \arg\max_{\tau} \sum_{i=1}^{N} x_i \ln(\alpha \mu_i^{(\tau)} + \lambda_0) - \alpha \sum_{i=1}^{N} \mu_i^{(\tau)}, \tag{1.4}$$

where $\hat{\tau}$ is the estimation of the TOA, λ is the signal defined in (1.2), and x is the received waveform.

If α is known, then the algorithm boils down to searching the received time window for the optimal τ. We present an efficient searching techniques in later chapters. However, if α is not known, it must generally be co-estimated. There is not a closed form solution for this co-estimation problem, so we employ an EM algorithm to iteratively solve for α:

$$\alpha^{(j+1)} = \frac{1}{\sum_{i=1}^{N} \mu_i^{(\tau)}} \left(\sum_{i=1}^{N} \frac{x_i \mu_i^{(\tau)} \alpha^{(j)}}{\alpha^{(j)} \mu_i^{(\tau)} + \lambda_0} \right). \tag{1.5}$$

The Cramér-Rao lower bound (CRLB), which is derived from the curvature of the log-likelihood function, gives us a theoretical lower bound on the variance of unbiased estimators. Although there is generally no guarantee that an ML estimator is unbiased, CRLBs

provide useful guidance.[2]

The CRLBs on τ and α for the Poisson case are

$$\text{var}(\tau) \geq \frac{\sum_{i=1}^{N} \frac{\left(\mu_i^{(\tau)}\right)^2}{\alpha\mu_i^{(\tau)}+\lambda_0}}{\alpha^2 \left(\sum_{i=1}^{N} \frac{\left(\partial\mu_i^{(\tau)}/\partial\tau\right)^2}{\alpha\mu_i^{(\tau)}+\lambda_0}\right)\left(\sum_{i=1}^{N} \frac{\left(\mu_i^{(\tau)}\right)^2}{\alpha\mu_i^{(\tau)}+\lambda_0}\right) - \alpha^2 \left(\sum_{i=1}^{N} \frac{\mu_i^{(\tau)}\left(\partial\mu_i^{(\tau)}/\partial\tau\right)}{\alpha\mu_i^{(\tau)}+\lambda_0}\right)^2} \quad (1.6)$$

and

$$\text{var}(\alpha) \geq \frac{\alpha^2 \sum_{i=1}^{N} \frac{\left(\partial\mu_i^{(\tau)}/\partial\tau\right)^2}{\alpha\mu_i^{(\tau)}+\lambda_0}}{\alpha^2 \left(\sum_{i=1}^{N} \frac{\left(\partial\mu_i^{(\tau)}/\partial\tau\right)^2}{\alpha\mu_i^{(\tau)}+\lambda_0}\right)\left(\sum_{i=1}^{N} \frac{\left(\mu_i^{(\tau)}\right)^2}{\alpha\mu_i^{(\tau)}+\lambda_0}\right) - \alpha^2 \left(\sum_{i=1}^{N} \frac{\mu_i^{(\tau)}\left(\partial\mu_i^{(\tau)}/\partial\tau\right)}{\alpha\mu_i^{(\tau)}+\lambda_0}\right)^2}. \quad (1.7)$$

Our suggested ad hoc method for censored Poisson data involves first estimating α using the EM algorithm for unsaturated data (1.5). We generate a λ_i profile using that estimated α according to (1.2), saturate that profile at the threshold, and then use the saturated profile in 1.4. This ad hoc technique provides a benchmark to see whether an EM algorithm that properly incorporates censored data is worth the additional conceptual and computational complexity.

Gaussian

Gaussian models generally have many attractive computation and analytical properties. The Poisson distribution is increasingly well approximated by the Gaussian distribution as the number of photon counts increases. Hence, we explore using a Gaussian approximation to the Poisson distribution. Since we model lightning signals as realizations of PPPs, we match the parameters of the Gaussian distribution to the mean and variance of the Poisson

[2]There are modifications of the CRLB that account for estimator bias, but these require derivatives of the bias, which are usually not available in practice.

distribution. The ML optimization program becomes

$$\hat{\tau} = \arg\max_{\tau} -\frac{1}{2}\sum_{i=1}^{N}\ln(\alpha\mu_i^{(\tau)}+\lambda_0) - \frac{1}{2}\sum_{i=1}^{N}\frac{\left(x_i-(\alpha\mu_i^{(\tau)}+\lambda_0)\right)^2}{\alpha\mu_i^{(\tau)}+\lambda_0}. \tag{1.8}$$

As in the true Poisson case, if α is unknown, it must be co-estimated. Since its ML estimate cannot be found in closed form, we developed an iterative EM algorithm:

$$\alpha^{(j+1)} = -\frac{N}{2\sum_{i=1}^{N}\mu_i^{(\tau)}} + \frac{1}{2}\sqrt{\frac{N^2}{\left(\sum_{i=1}^{N}\mu_i^{(\tau)}\right)^2} + \frac{4\alpha^{(j)}}{\sum_{i=1}^{N}\mu_i^{(\tau)}}\Xi}, \tag{1.9}$$

where

$$\Xi = N - \alpha^{(j)}\sum_{i=1}^{N}\frac{\mu_i^{(\tau)}}{\alpha^{(j)}\mu_i^{(\tau)}+\lambda_0} + \alpha^{(j)}\sum_{i=1}^{N}\frac{\mu_i^{(\tau)}x_i^2}{\left(\alpha^{(j)}\mu_i^{(\tau)}+\lambda_0\right)^2}. \tag{1.10}$$

The CRLB of τ for the Gaussian model is

$$\text{var}(\tau) \geq \frac{\frac{1}{2}\sum_{i=1}^{N}\frac{\left(\mu_i^{(\tau)}\right)^2}{(\alpha\mu_i^{(\tau)}+\lambda_0)^2} + \sum_{i=1}^{N}\frac{\left(\mu_i^{(\tau)}\right)^2}{\alpha\mu_i^{(\tau)}+\lambda_0}}{\beta}, \tag{1.11}$$

where

$$\begin{aligned}\beta = &\left(\frac{\alpha^2}{2}\sum_{i=1}^{N}\frac{\left(\partial\mu_i^{(\tau)}/\partial\tau\right)^2}{(\alpha\mu_i^{(\tau)}+\lambda_0)^2} + \alpha^2\sum_{i=1}^{N}\frac{\left(\partial\mu_i^{(\tau)}/\partial\tau\right)^2}{\alpha\mu_i^{(\tau)}+\lambda_0}\right)\left(\frac{1}{2}\sum_{i=1}^{N}\frac{\left(\mu_i^{(\tau)}\right)^2}{(\alpha\mu_i^{(\tau)}+\lambda_0)^2} + \sum_{i=1}^{N}\frac{\left(\mu_i^{(\tau)}\right)^2}{\alpha\mu_i^{(\tau)}+\lambda_0}\right)\\ &-\left(\frac{\alpha}{2}\sum_{i=1}^{N}\frac{\mu_i^{(\tau)}(\partial\mu_i^{(\tau)}/\partial\tau)}{(\alpha\mu_i^{(\tau)}+\lambda_0)^2} + \alpha\sum_{i=1}^{N}\frac{\mu_i^{(\tau)}(\partial\mu_i^{(\tau)}/\partial\tau)}{\alpha\mu_i^{(\tau)}+\lambda_0}\right)^2.\end{aligned} \tag{1.12}$$

The CRLB of α for the Gaussian model is

$$\text{var}(\alpha) \geq \frac{\frac{\alpha^2}{2}\sum_{i=1}^{N}\frac{\left(\partial\mu_i^{(\tau)}/\partial\tau\right)^2}{(\alpha\mu_i^{(\tau)}+\lambda_0)^2} + \alpha^2\sum_{i=1}^{N}\frac{\left(\partial\mu_i^{(\tau)}/\partial\tau\right)^2}{\alpha\mu_i^{(\tau)}+\lambda_0}}{\beta}, \tag{1.13}$$

where β is defined in (1.12).

Our proposed Gaussian Ad Hoc runs along the same lines as the previously suggested Poisson Ad Hoc method, and provides the same kind of benchmark.

The derivations, implementations, and comparisons of these methods are provided in later chapters.

Censored Models

The aforementioned algorithms do not properly handle cases where the received signal has undergone saturation. Our research focuses on correctly incorporating saturation effects into ML models by invoking censored versions of the previously mentioned probability models. In our censored models, for all received data less than a specified maximum value, the usual probability model (Poisson or Gaussian) is followed. However, values at or above the specified threshold are grouped into one distinct bin with probability given by the discrete sum (or continuous integral, as appropriate) of the usual probability model from the threshold to infinity. This is equivalent to one minus the cumulative distribution function (CDF) evaluated at the specified maximum value.

Our censored models all follow the same optimization program structure:

$$\hat{\tau} = \arg\max_{\tau} \ \ell(\theta|x) = \sum_{i=1}^{N} g_i(t), \tag{1.14}$$

where $\theta = [\tau \ \ \alpha]^T$. The only difference lies in $g(t)$.

For the censored Poisson model,

$$g_i(t) = \begin{cases} \left(x_i \ln(\alpha\mu_i^{(\tau)} + \lambda_0) - \alpha\mu_i^{(\tau)} - \lambda_0 - \ln(x_i!)\right) & x_i < T \\ \ln\left(1 - \exp\left\{-(\alpha\mu_i^{(\tau)} + \lambda_0)\right\} \sum_{j=0}^{\lfloor T \rfloor} (\alpha\mu_i^{(\tau)} + \lambda_0)^j / j!\right) & x_i \geq T \end{cases}, \tag{1.15}$$

and for the censored Gaussian model

$$g_i(t) = \begin{cases} -\frac{1}{2}\ln\left(\alpha\mu_i^{(\tau)} + \lambda_0\right) - \frac{\left(x_i - (\alpha\mu_i^{(\tau)} + \lambda_0)\right)^2}{2\left(\alpha\mu_i^{(\tau)} + \lambda_0\right)} & x_i < T \\ \ln\left(1 - \frac{1}{\sqrt{2\pi\left(\alpha\mu_i^{(\tau)} + \lambda_0\right)}} \int_{-\infty}^{T} \exp\left\{ -\frac{\left[t - \left(\alpha\mu_i^{(\tau)} + \lambda_0\right)\right]^2}{2\left(\alpha\mu_i^{(\tau)} + \lambda_0\right)} \right\} dt \right) & x_i \geq T \end{cases}. \quad (1.16)$$

For the above loglikelihoods, T represents the specified maximum value of the sensor. The saturated signal x will never actually be greater than T.

We have developed EM algorithms for both the Poisson and Gaussian censored models to estimate the unknown scaling factor, α.

For the censored Poisson model, we have

$$\alpha^{(j+1)} = \frac{C}{\sum_{i=1}^{N} \mu_i^{(\tau)}}, \quad (1.17)$$

where

$$C = \sum_{i \in \mathcal{N}} \frac{x_i \alpha^{(j)} \mu_i^{(\tau)}}{\alpha^{(j)} \mu_i^{(\tau)} + \lambda_0} + \sum_{i \in \mathcal{C}} \left(\sum_{y_i^S = 0}^{\infty} y_i^S \cdot \frac{\left[e^{-\alpha^{(j)} \mu_i^{(\tau)}} \left(\alpha^{(j)} \mu_i^{(\tau)} \right)^{y_i^S} / y_i^S! \right] \left[1 - F(T - y_i^S - 1; \lambda_0) \right]}{1 - F\left(T - 1; \alpha^{(j)} \mu_i^{(\tau)} + \lambda_0 \right)} \right), \quad (1.18)$$

where $F(A; \lambda)$ is the CDF of a Poisson distribution evaluated at A, defined by the intensity λ, and T is the censoring threshold.

For the censored Gaussian model, we have

$$\alpha^{(j+1)} = \frac{-N + \sqrt{N^2 + 4C \sum_{i=1}^{N} \mu_i^{(\tau)}}}{2 \sum_{i=1}^{N} \mu_i^{(\tau)}}, \quad (1.19)$$

where

$$C = \sum_{i \in \mathcal{N}} \left(\alpha^{(j)} - \frac{(\alpha^{(j)})^2 \mu_i^{(\tau)}}{\alpha^{(j)} \mu_i^{(\tau)} + \lambda_0} + \frac{(\alpha^{(j)} x_i)^2 \mu_i^{(\tau)}}{(\alpha^{(j)} \mu_i^{(\tau)} + \lambda_0)^2} \right)$$
$$+ \sum_{i \in \mathcal{C}} \left(\int_0^{\infty} \frac{(y_i^S)^2}{\mu_i^{(\tau)}} \cdot \frac{\left[\exp\left\{ -\frac{\left(y_i^S - \alpha^{(j)} \mu_i^{(\tau)}\right)^2}{2\alpha^{(j)} \mu_i^{(\tau)}} \right\} \Big/ \sqrt{2\pi \alpha^{(j)} \mu_i^{(\tau)}} \right] \left[1 - F(T - y_i^S; \lambda_0, \lambda_0) \right]}{1 - F(T, \alpha^{(j)} \mu_i^{(\tau)} + \lambda_0, \alpha^{(j)} \mu_i^{(\tau)} + \lambda_0)} dy_i^S \right), \tag{1.20}$$

where $F(A; \mu, \sigma^2)$ is the Gaussian CDF with mean μ and variance σ^2 evaluated at A, and T is the saturation threshold.

Derivations, implementations, and comparisons of these algorithms are contained in later chapters.

1.3 Organization of the Thesis

Chapter 2 outlines the machine learning techniques employed to filter the FORTÉ database and generate the lightning templates. Chapter 3 derives the TOA estimation algorithms, the scaling factor estimation algorithms, and the CRLBs for the uncensored probability distributions. It also contains results from several MC simulations comparing these methods. Chapter 4 contains the derivation and implementation of the censored estimation algorithms for the scaling factor α. Chapter 5 describes a method to estimate the CRLBs of the censored models using the EM framework outlined in Chapter 4. Finally, Chapter 6 contains the results of several MC simulations comparing all models and offers some conclusions.

CHAPTER 2
LIGHTNING TEMPLATE GENERATION

2.1 Introduction

About 100 lightning strikes occur every second [12]. This chapter uses data collected by the FORTÉ satellite, a small satellite jointly launched by Las Alamos National Laboratory (LANL) and Sandia National Laboratories (SNL) in 1997 to collect optical and radio-frequency transient signals from lightning. A photodiode detector (PDD) records the optical signal with a sampling interval of 15 microseconds (μsec). Each lightning intensity profile analyzed in this chapter consists of 128 samples totalling 1.92 milliseconds (msec)[12]. This chapter analyzes $914\,894$ signals from the year 2001.

As seen in Fig. 2.1, the measured signals include noise-like waveforms, ramps, particle excitations (signals too sharp to be lightning strikes, caused by high energy particles striking the PDD), various shapes of lightning intensity profiles, and other phenomena.

This chapter proposes a machine learning approach to characterizing lighting strikes and also differentiating between actual lightning strikes recorded by the PDD and mistakenly collected noise, providing an alternative to manually inspecting the data[16]. Our algorithm classified $72\,096$ of the $914\,894$ signals (about 7.9%) as "non-lightning," which includes noise signals, particle excitation events, DC readings, ramps, and other miscellaneous signals.

Once the non-lightning profiles are removed from the data set, we create familial clusters of lightning signals, as well as representative signals for each family. These results may assist with the development of statistical models of lightning occurrence, in addition to allowing for efficient sampling of the space of lightning signatures. We found nine families of lightning and sampled $18\,000$ signals based off of the probabilistic distribution of

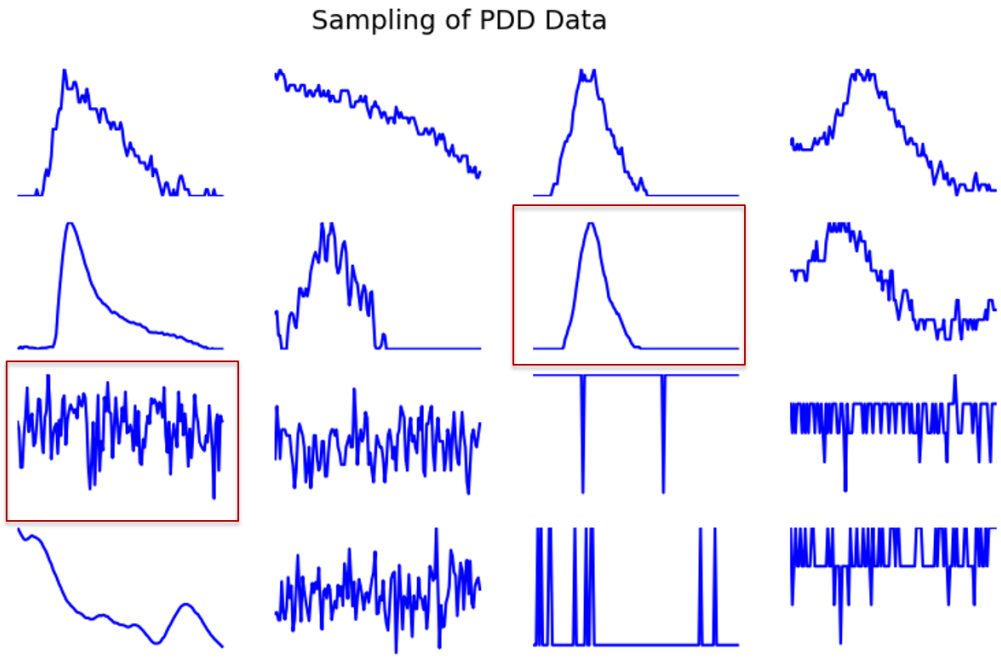

Figure 2.1: Examples of signals from 2001 FORTÉ PDD data. Highlighted signals will be examined later on.

which family each signal belongs to. Finally, the 1000 signals in the entire lighting-only data set that are furthest from their family's mean are selected to form an anomaly data base for further analysis.

The FORTÉ PDD records signals in the units of *counts* that relate to the photocurrent through a piecewise linear transfer function. Its three-stage gain compression supports larger dynamic range than a simple linear conversion. The photocurrent is proportional to the optical irradiance, and is recovered from the count data by passing the signal through the functional inverse of the compression. This data, called the *irradiance* signal, is reported in units of irradiance (W/cm^2). We used the counts data for eliminating non-lightning signatures, and the irradiance representation in all other analysis.

This chapter consists of three main parts. Section 2.2 consists of a brief overview of machine learning and its application in this chapter. Section 2.3 discusses methods and results for culling most of the signals from the database. Section 2.4 explains the methods, results, and initial analysis of creating familial clusters of lightning.

2.2 Machine Learning

This section gives a brief overview of machine learning and its applications in this chapter. To establish notation, the inputs to the algorithms $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ are called "feature vectors." Each member of a feature vector is referred to as a "feature." These feature vectors do not have to correspond to the 128-dimensional time-domain signals from the database. Instead, the PDD records can be processed through an application-specific "feature vector transform" that represents the same signals in a different way.

In this chapter, unsupervised machine learning (UML) is employed to find structure in a given data set. Training data for a UML algorithm consists of a set of input feature vectors, i.e. a set of waveforms passed through the feature vector transform, without any sort of labeling or any assumptions from the implementor (other than assuming all feature vectors have the same number of samples per waveform). The UML algorithm looks for groups of waveforms that are most similar to each other, according to some application-dependent similarity criterion.

In contrast to UML, supervised machine learning (SML) requires *labeled* data. Supervised machine learning assigns an input feature vector to an output, like a category, based on a learned function. This chapter is concerned with classification, which maps each input feature vector to a category chosen from a list of categories. To separate signals into either *lightning* or *non-lightning* is a binary classification task. To train and test the algorithm, we created a *tagged* data set from a subset of the original data. We extracted a subset of the data and hand-labeled each waveform as either lightning or non-lightning. We used this subset to test the SML algorithm performance [17].

2.3 Separation of Non-Lightning Profiles

This section explains the derivation of and results of our technique for separating lightning profiles from non-lightning profiles. Here, the goal is to identify and remove "noise" signals

that do not contain actual lightning signatures, not "denoise" a particular recorded signature that is a mixture of true lightning and noise effects.

2.3.1 Approach

The process for culling non-lightning signals is explained briefly here; each step is explained in more detail below. We created two data sets, a *testing* set and a *training* set. Each record in the testing subset was hand-labeled as either lightning or non-lightning. The training data was processed through the feature vector transform, and then a UML algorithm grouped the feature vectors into ten clusters, and each group was inspected and labeled (as a whole group) as either lightning or non-lightning. The clustering model, learned by the algorithm, along with the post-clustering labeling, constitute a binary classification algorithm. Any waveform passed through the feature vector transform and then clustered is labeled as lightning or non-lightning by the algorithm.

Applying the binary classification algorithm to the testing data characterizes the effectiveness of the algorithm. We finally used the derived algorithm to cull the entire original data set.

The 2001 FORTÉ PDD data contains $914\,894$ signals. We sequestered 880 signals for testing purposes. 440 of these records were hand labeled as lightning, and 440 were hand-labeled as non-lightning. The lightning subset contained a wide variety of observed lightning shapes (including low SNR signals), while the non-lighting signals contained examples of noise, particle excitation, ramps, and other noise-like signals. Figure 2.2 shows examples from the test set.

After manually examining many lightning and non-lightning signals, it became apparent that two kinds had noticeably varying frequency content. The non-lightning signals consisted of mostly high frequency content, almost strictly DC frequency content, or a few time-domain impulses. Therefore, we chose a feature vector transformation so that our algorithm interpreted signatures in the frequency domain, rather than the time domain.

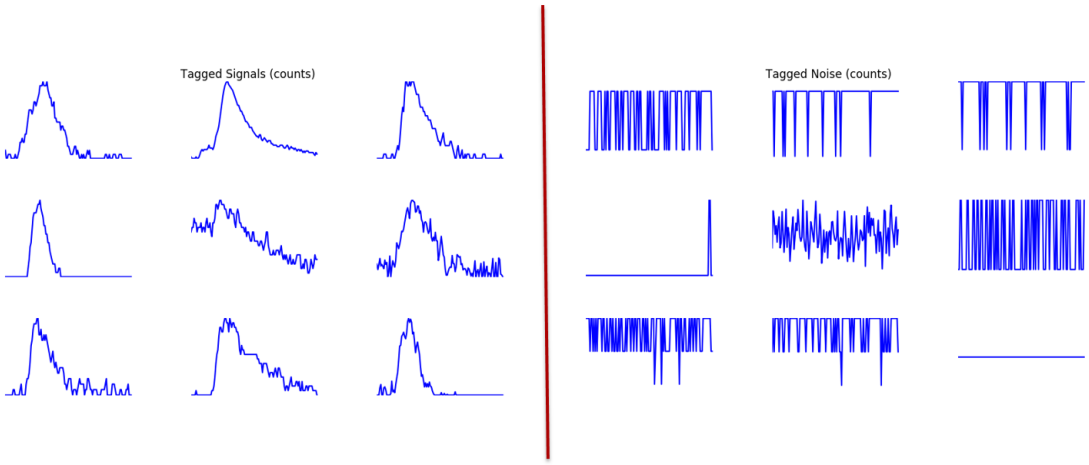

Figure 2.2: Examples of tagged test set. We sequestered 880 signals from the original data set, half of which were lightning signals, and half of which were non-lightning signals.

After experimenting with various truncations of the discrete cosine transform (DCT) and FFT, we settled on feature vectors consisting of the absolute values of the first 25 positive-frequency coefficients of the 128 point FFT.

Figures 2.3 and 2.4 show an example of the feature vectors derived from the signals outlined by boxes in Fig. 2.1. In this example, the non-lightning signature has a significantly higher DC value, as well as frequency content across all coefficients. The lightning signal, on the other hand, has less magnitude at all coefficients, and the vast majority of the energy of the signal is contained in the first 10 coefficients.

Once these FFT-based feature vectors are created, we force each individual feature to have zero mean and an absolute maximum of one across the entire data set, as is customary. These offset and scaling factors used in this step must be remembered, since the algorithm expects all new signals that may be classified later on to have feature vectors modified by the same factors.

Once the data has been transformed and preprocessed, it is ready to be clustered. For separating non-lightning signatures from lightning signatures, we used Gaussian mixture models (GMMs), which are linear superpositions of Gaussian probability density functions. After trying mixtures ranging from five to fifteen components, we decided upon ten. When

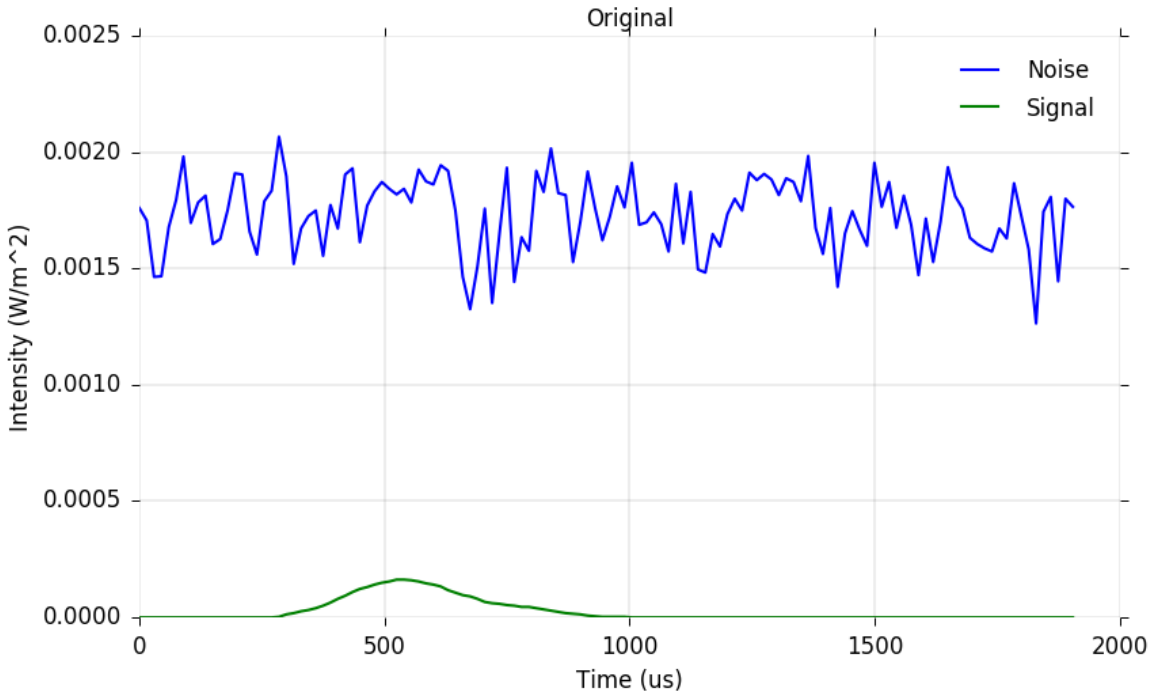

Figure 2.3: Original signals showing one lightning signal and one "noise", or non-lightning signal. The feature vector transform of each of these signals is plotted in the next figure.

we tried more than ten, some groups would hardly have any signatures assigned to them, and less than ten tended to underperform in cross-validation tests. As is customary with GMMs, we iterate an Expectation-Maximization algorithm to maximize the loglikelihood of each feature vector being assigned to one of the ten Gaussians [17].

The GMM algorithm clustered the training data set into ten different groups. Nine randomly chosen examples from each cluster are shown in Figs. 2.5 and 2.6. To transition from unsupervised clustering to supervised classification, we manually inspected these figures and hand-labeled some of the groups as non-lightning; these non-lightning groups are shown in boxes.

2.3.2 Results

This section discusses the results of our separation methodology. We first describe the performance of the classification algorithm on the test data set. We then describe the result

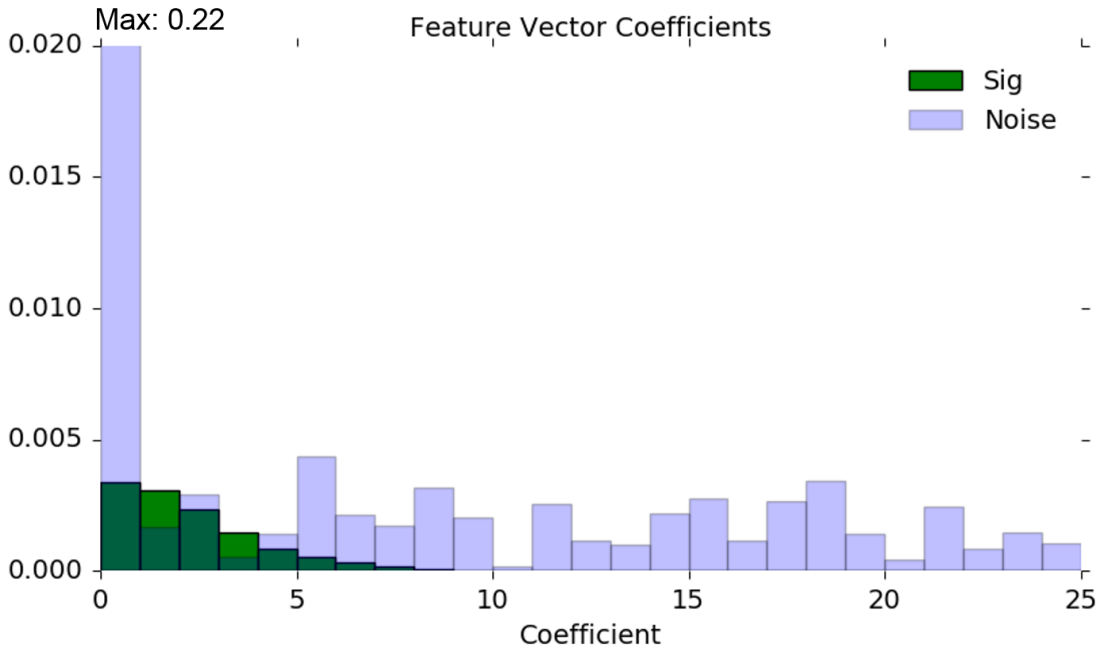

Figure 2.4: Feature vectors of the original signals plotted in Fig. 2.3 after going through the feature vector transform of selecting the first 25 positive FFT coefficients of a 128 point FFT.

of applying the classification algorithm to the entire data set.

Thirteen of the 880 tagged waveforms were misclassified, giving an "official" misclassification rate of 1.7%. Figure 2.7 shows the misclassified signals. Inspecting those signals, it is clear that human understanding and classification is as much a factor as the classification algorithm itself. For example, the first two signals "misclassified" by the algorithm are ramps. We had previously manually tagged these as lightning, since there appears to be structure in the signals. However, the signals are quite different than a typical lightning intensity profile. Perhaps they are actually non-lightning profiles, or should be treated as "noise" regardless of their cause. The remaining 11 were manually tagged as non-lightning, but "misclassified" by the algorithm as lightning. However, many of these 11 signals show signs of structure; perhaps they represent far-away lightning strikes exhibiting quantization artifacts. The nature of lightning vs. non-lightning signals is not always clear and well-defined. This issue of variations in human interpretation arise in evaluating the per-

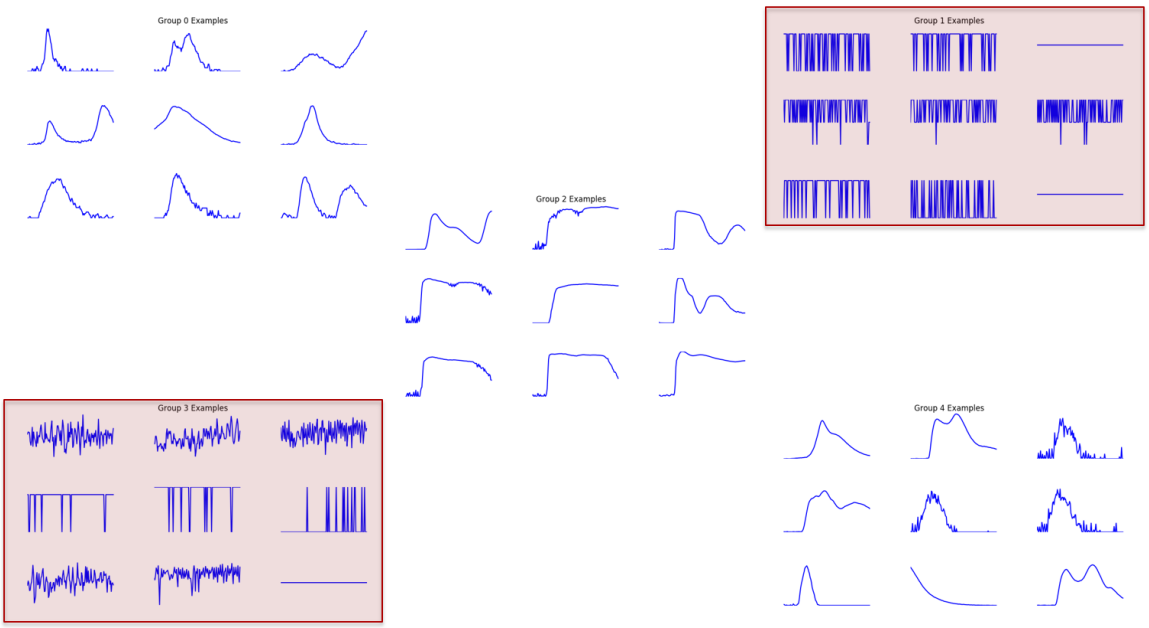

Figure 2.5: Random samplings of Groups 0-4 from the GMM clustering of the entire data set using the FFT feature vector transform.

formance of any supervised algorithm, motivating the use of the term "official" earlier in this paragraph. These structured "non-lightning" signals play an important role later in this chapter, because they are not culled after the application of the algorithm in this section.

Application of the classification method to the entire data set declared 38 269 records to be non-lightning (4.2%). These signals were removed from the database, leaving 876 625 signals for the familial cluster analysis in Section 2.4.

2.3.3 Alternative methods

We selected the magnitude of the first 25 coefficients of the FFT as a feature vector, with GMM as a clustering method, based on the resulting performance. This section mentions some other methods we experimented with but ultimately rejected.

Other feature vectors we tried included the original time-domain signals, low-pass filtered versions of the time-domain signals, the first n coefficients of the DCT, nonnegative matrix factorization basis expansion coefficients, and dictionary-learning basis expansion coefficients. FFT coefficients generated the least number of misclassifications of the test

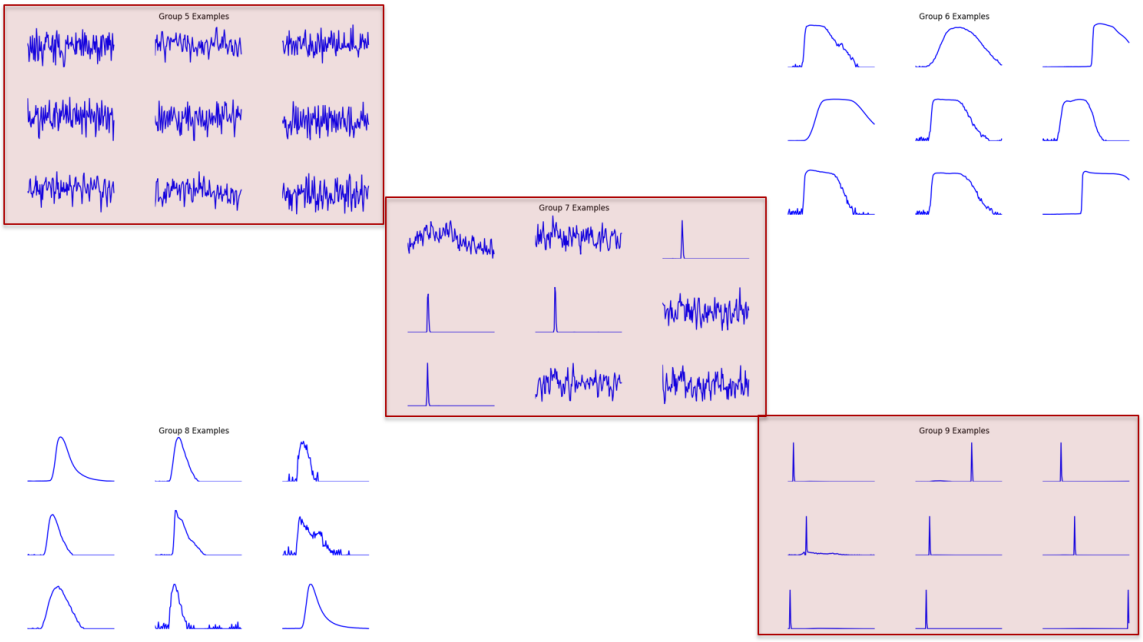

Figure 2.6: Random samplings of Groups 5-9 from the GMM clustering of the entire data set using the FFT feature vector transform.

data. Furthermore, the resulting "errors" occurred for signatures that were the most difficult for a human to adjudicate, suggesting that the feature vector was performing about as well as a human.

Although we ultimately chose GMM for clustering, we also tried k-means clustering[17]. However, the lightning and non-lightning separation appears to be non-spherical, and k-means produced rather poor results. We also investigated a manifold clustering method called spectral clustering; unfortunately, it could only be successfully completed on 1% of the data because of its tremendous memory requirements.

We also trained and tested a Support Vector Machine on the data set. While it performed similarly to our GMM-based clustering method, its computational complexity was larger. Support Vector Machines are not as scalable as the "cluster then classify" approach described in Section 2.3.1.

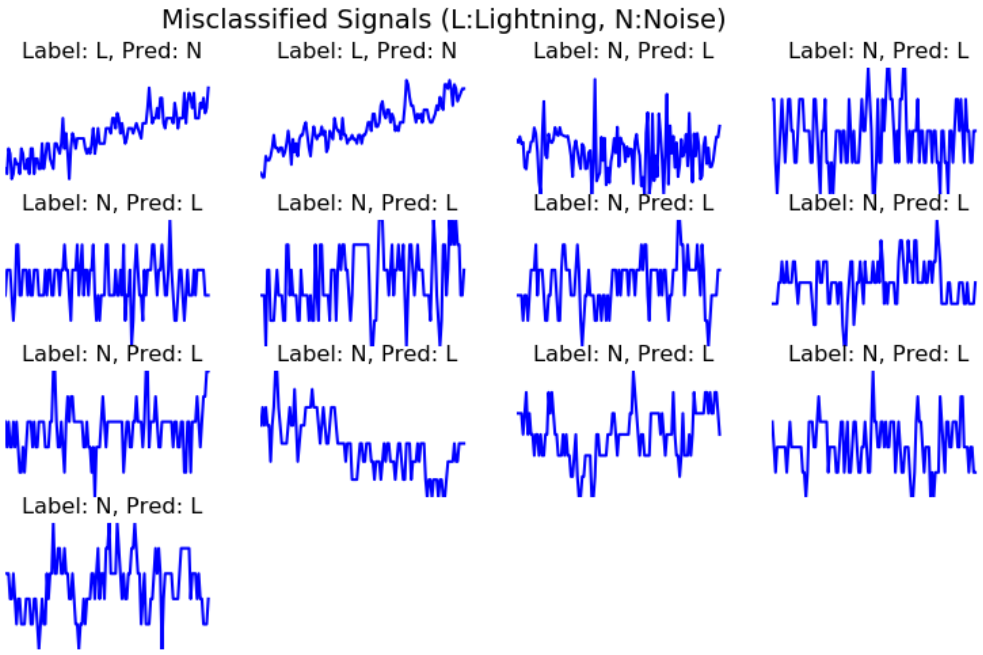

Figure 2.7: Misclassified signals using our trained classification algorithm on the testing data set of 440 signals. As can be seen, the human interpretation of lightning vs. non-lightning signals also plays a role. The "official" misclassification rate of our trained classifier is 1.7%.

2.4 Identifying Lightning Families

This section discusses the methodology and results of familial cluster generation.

Figure 2.8 plots the two most significant components of a principal component analysis (PCA) decomposition of the data the remains after the culling stage of Section 2.3. The PCA data has no obvious clusters when projected onto these two dimensions. Instead, there is a large cluster in the middle, with outliers on the fringe. Hence, we do not expect there to be obvious familial clusters with low intra-cluster variance.

2.4.1 Approach

To generate familial clusters, we create a basis for the irradiance data using nonnegative matrix factorization (NNMF). Each signal is projected onto this basis, and the projection

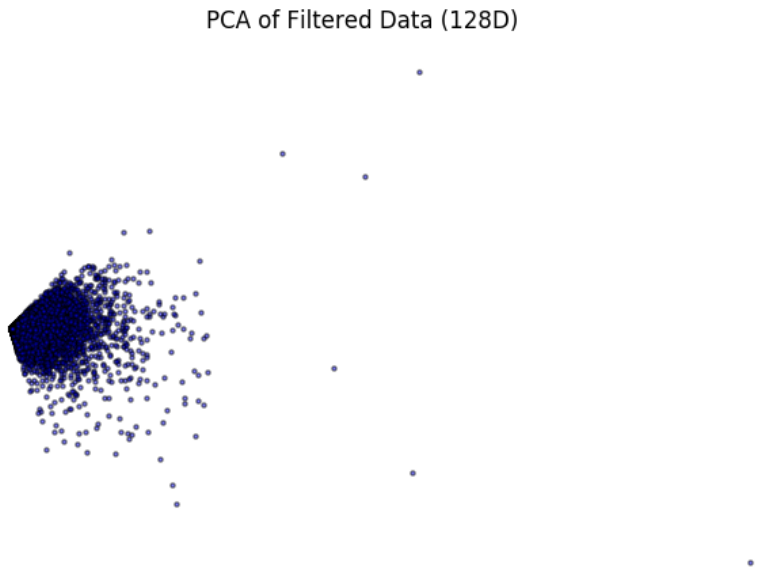

Figure 2.8: PCA of time-domain signals. With this two-dimensional representation of the lightning data, we do not expect there to be obvious familial clusters with low intra-cluster variance.

coefficients are used as feature vectors for a clustering algorithm.

NNMF approximates a matrix as a linear sum of nonnegative components, formerly solving the optimization

$$\min_{\boldsymbol{B}\in\mathbb{R}^{d\times r},\boldsymbol{C}\in\mathbb{R}^{r\times n}} ||\boldsymbol{X}-\boldsymbol{B}\boldsymbol{C}||_F^2,\ \boldsymbol{B}\geq\boldsymbol{0},\ \boldsymbol{C}\geq\boldsymbol{0}, \tag{2.1}$$

where the $\geq \mathbf{0}$ notation indicates that all elements of a matrix are nonnegative.

If the columns of $\boldsymbol{X}$ are waveforms, the columns of $\boldsymbol{B}$ contain the r most important nonnegative basis vectors, and $\boldsymbol{C}$ contains the expansion coefficients of the n waveforms in $\boldsymbol{X}$ in terms of these r basis vectors [18].

In this section, we first normalize each irradiance profile to have unit L_2 norm. (Other norms, such as L_1 or L_∞, could be used). Figure 2.9 shows the $r = 25$ basis waveforms (columns of $\boldsymbol{B}$) resulting from the $876\,625$ lightning profiles that remained after the culling

from Section 2.3.

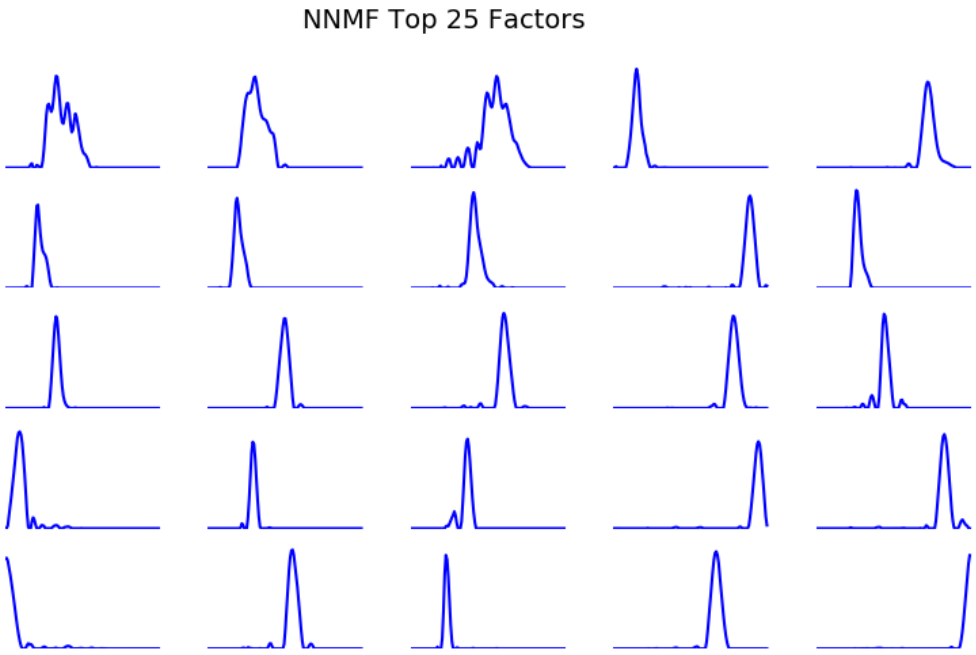

Figure 2.9: Nonnegative matrix factorization basis waveforms generated via decomposition of all lightning signals.

Once a NNMF basis set is created, generating feature vectors for new waveforms involves normalizing the waveforms to have unit L_2 norm, followed by projecting the normalized waveforms onto the basis set. For our existing normalized waveforms in $\boldsymbol{X}$, these expansion coefficients are found in $\boldsymbol{C}$.

Our feature vector consists of 25 projection coefficients. Figure 2.9 shows that the basis set consists of Gaussian-like bumps with peaks at various times. After the NNMF feature vector transform, we force each feature to have zero mean and unit absolute maximum, like we did with the FFT feature vectors in Section 2.3.1.

Figure 2.10 plots two example time-domain waveforms (before L_2 normalization). Figure 2.11 shows the corresponding transformed feature vectors computed *after* L_2 normalization and basis projection, but *before* the processing step that forces zero mean and unit absolute maximum features (to make it easier to compare these two particular examples).

The shape and time-shift differences are clearly visible in the NNMF domain.

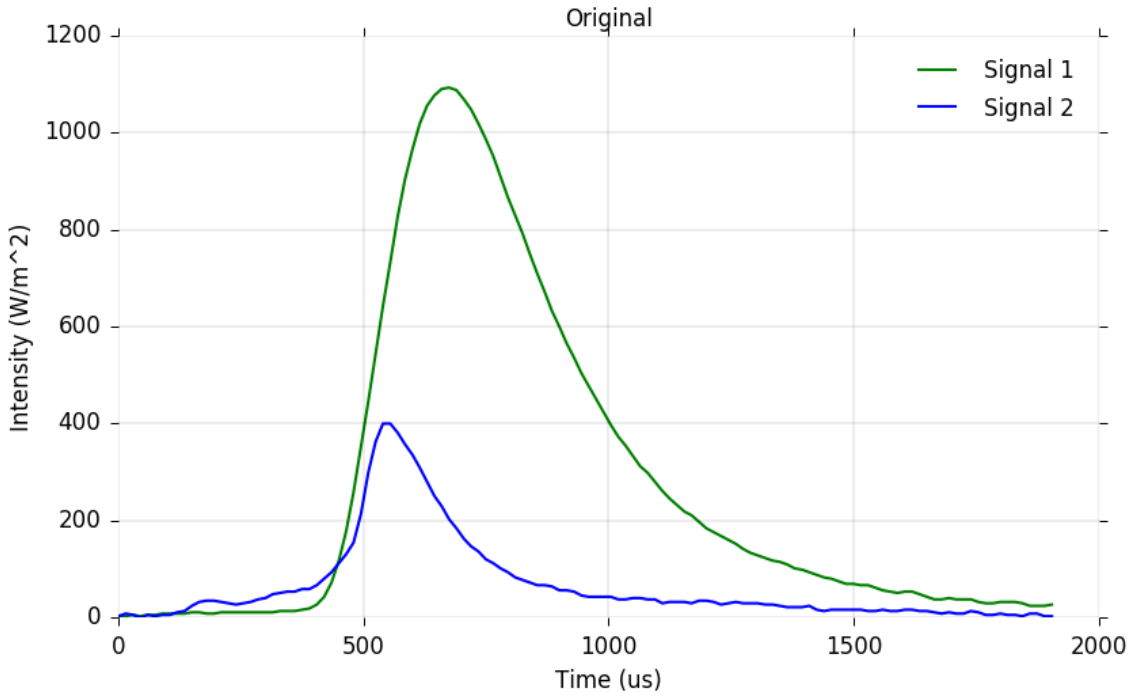

Figure 2.10: Original signals showing two different lightning signals. The NNMF feature transform of each of these signals is plotted in the following figure.

Compare the PCA analysis of the time-domain lightning signals in Fig. 2.8 to the PCA analysis of the normalized (zero-mean, unit absolute maximum) NNMF feature vectors shown in Figure 2.12. While the signals still do not show evident clusters, they are more uniformly and widely spaced, which can improve clustering.

We clustered the lightning waveforms using the NNMF feature vectors into ten clusters using a new GMM. These clusters are fundamentally different than those in the lightning/non-lightning separation of Section 2.3, since they are most directly based on temporal shape, rather than frequency content.

2.4.2 Results

This section discusses the results of familial clustering, beginning with the representation of signal families by their means and covariances. We then analyze each individual group. We

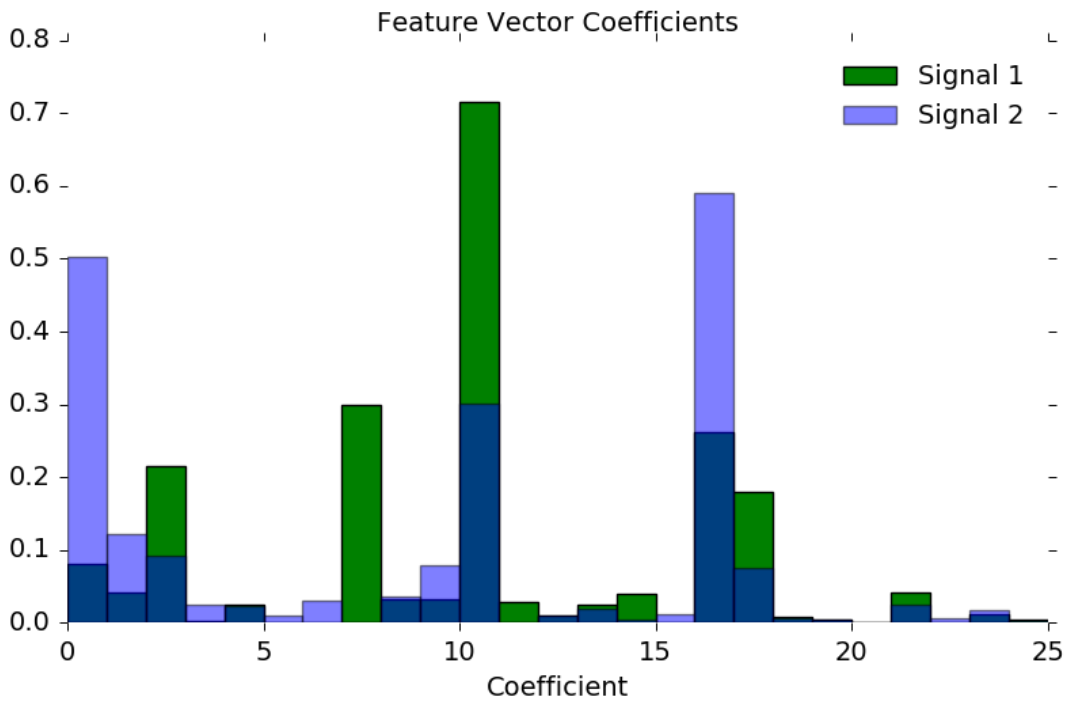

Figure 2.11: Feature vectors of the signals from Fig. 2.10. The differences of the two signal are obvious in the new domain.

also describe how, independent of mean-and-covariance representations, we capture a set of $18\,000$ signals that describes the families and a set of outliers. Finally, we demonstrate a lightning signal synthesis method.

As explained in the previous subsection, ten groups were generated by the GMM clustering method. However, only nine of the groups are useful for familial analysis and lightning synthesis. As mentioned in Section 2.3.2, the original culling stage misclassified some noisy signals with non-white structure as lightning. These signals are mostly useless. Fortunately, they tend to group together as a result of familial clustering. Therefore, the familial group containing the useless signals is removed in the rest of our analysis, forming a second stage of culling. This useless group contains $33\,827$ signals (3.7%) which, together with the first stage's culling of 4.2%, results in a total "non-lightning" percentage of 7.9%.

We averaged the original time-domain signals (before L_2 normalization) in each cluster to form representative signals, which are shown in Fig. 2.13. Group 1 contains the $33\,827$

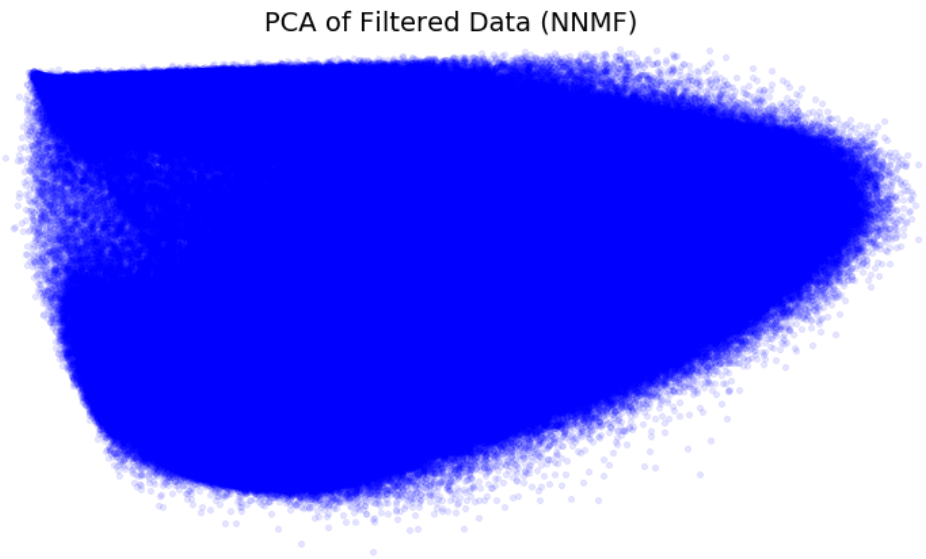

Figure 2.12: PCA of NNMF feature vectors. The data is much more uniformly and widely spaced in this new basis.

newly-discovered non-lightning signatures. The 9 remaining distinct families are labeled Group 0 and Groups 2 through 9. Figure 2.14 displays how many signals are in each group. Group 5, by far the largest group, contains 35% of the signals recorded.

Figure 2.15 is the same figure as Fig. 2.12, but with group assignments labeled by color. The NNMF feature vector transformation spreads out the input space, allowing for sections to be carved into clusters.

In the remainder of this chapter, we show means and covariances of the original unnormalized time-domain signals of various groups. Figure 2.16 shows this information for Group 0; the remaining groups can be found in Appendix A. The covariance images show the absolute value of the covariance entries on a log scale. The portions of the plot that are yellow (or simply lighter in greyscale) indicate parts of the lightning intensity profile that vary widely throughout the family. Group 0 has small variance at the beginning, increasing throughout the hump of the profile. The mean and variance of Group 0 can be described

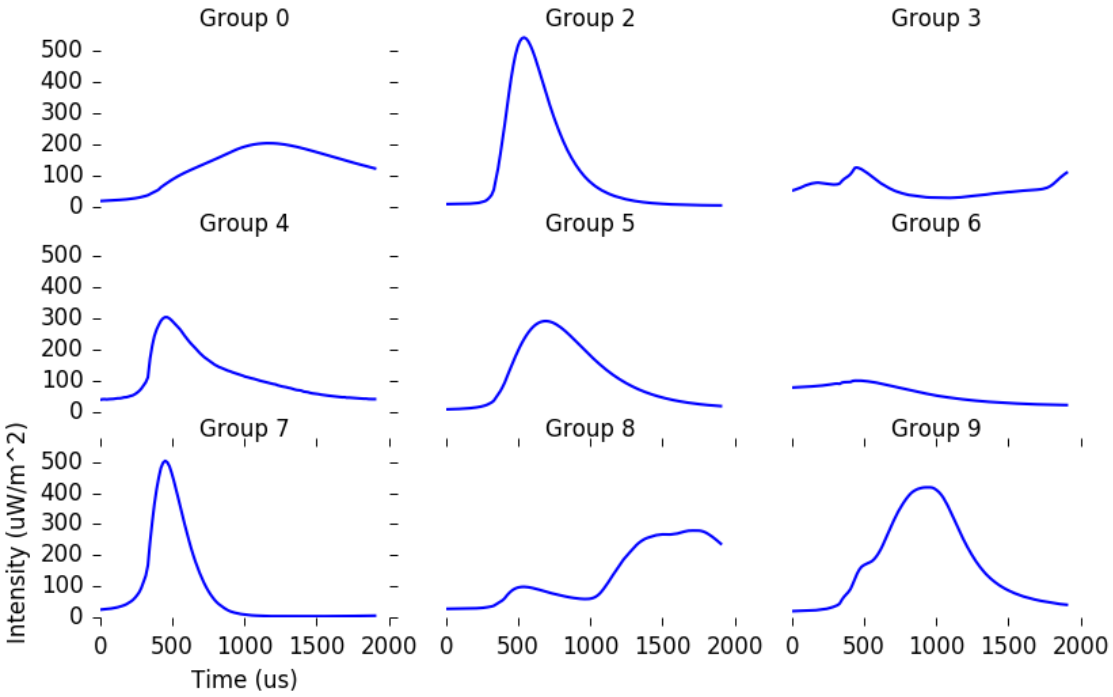

Figure 2.13: Group representative signals generated by computing the average of the time-domain signals in each group.

as starting low and flat, but having a wide hump starting around sample 20 (300µsec). Different time samples are highly correlated. Other groups have their own characteristics. For example, Group 7 is defined by signals with flat beginnings and ends and an extremely sharp hump.

To build a set of signals that cover the entire data space, we break each cluster into ten deciles based on the loglikelihood of the GMM. Figure 2.17 shows a histogram of this breakdown for Group 0; the rest of the groups can be found in Appendix B. The first signal in each decile is plotted next to its label in the legend. A higher loglikelihood indicates that the clustering algorithm is more confident that this waveform resides in this group and believes it is more like the representative signal. Decile 1, the nearest-to-mean 10% of Group 0, lies on the far right side of the graph. The exemplar of Decile 1, plotted in the legend, is similar to the representative signal of Group 0, which is plotted in the top center of the figure. The exemplar from Decile 2 still resembles the Group 0 representative, but as

Group	Signals	%
0	102,353	12
1	33,827	4
2	171,235	20
3	11,961	1
4	24,077	3
5	302,527	35
6	97,406	11
7	98,330	11
8	19,184	2
9	15,725	2
Total	876,625	

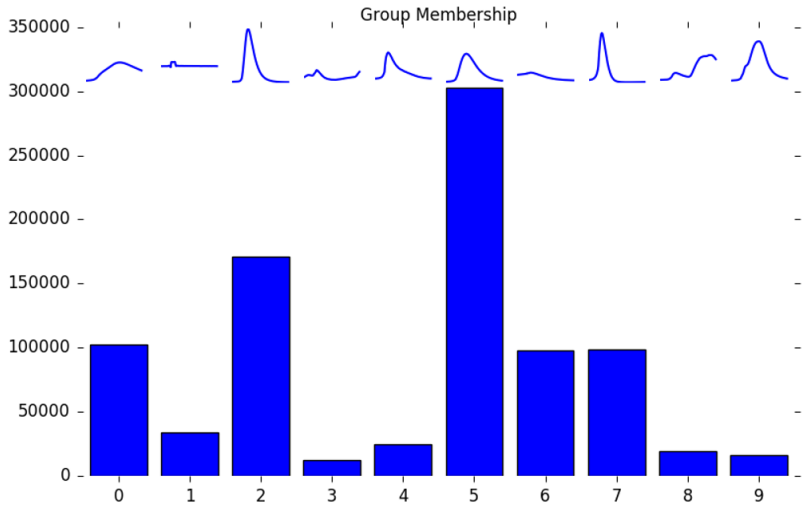

Figure 2.14: Data set broken down by group. Certain groups are more likely to occur (i.e. Group 5) than others (i.e. Group 3).

the deciles increase, the signals look less and less like group representative. As expected, this trend is consistent for all of the groups. The more dissimilar a signal is to the group's centroid, the smaller the likelihood is that the signal falls in that group.

The signals somewhat resemble each other when moving from decile to adjacent decile, but may vary drastically between distant deciles. For example, Deciles 9 and 10 are similar, but Deciles 1 and 10 are not. This is indicative that, although signals in Decile 10 are not obviously similar to the representative signal of Group 0, they are similar to other signals in the group.

To create a subset of $18\,000$ signals that reasonably represents the space of lightning signatures, we take the 200 highest-loglikelihood signals from each of the ten deciles of each of the nine groups. Including samples from each decile mitigates large intra-cluster variation. We also created an "anomalous set" consisting of the 1000 signals that have the lowest loglikelihoods of the entire lightning data set. Figure 2.18 shows a random selection of these anomalous signals.

Given the time-domain mean and covariance matrix of a family of lightning signals, we can synthesize new lightning profiles as realizations of the corresponding Gaussian

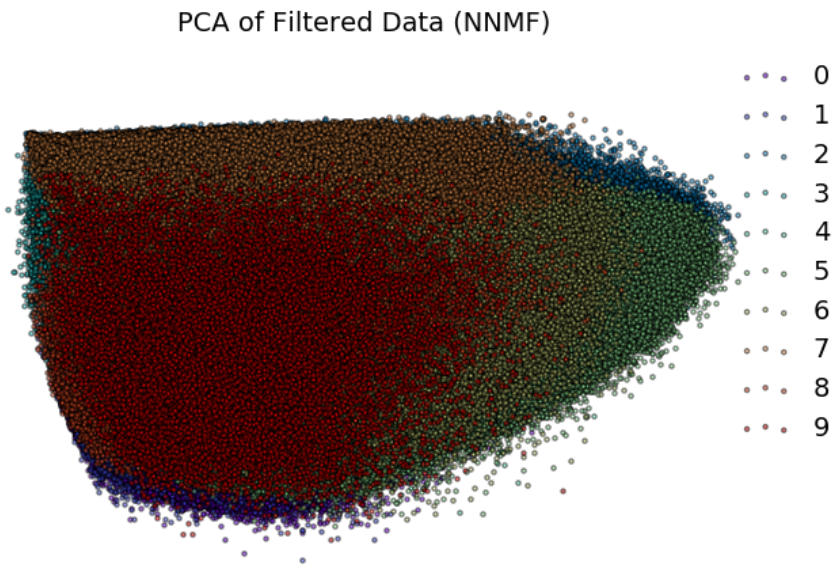

Figure 2.15: Labeled PCA of NNMF feature vectors. The NNMF feature space allows for sections of the data to be carved into clusters.

multivariate density.[1]

Figure 2.19 shows a lightning signal randomly selected from Group 0 and a signal synthesized using Group 0's mean and covariance matrix. Occasionally, this synthesis method can create intensity profiles with a sharp, unnatural roll-off (see Appendix C). Enforcing some sort of smoothness requirement might address this problem; we leave this as an avenue for future work.

[1] In practice, it may be necessary to truncate these Gaussian-based signatures at zero, since real lightning profiles are nonnegative.

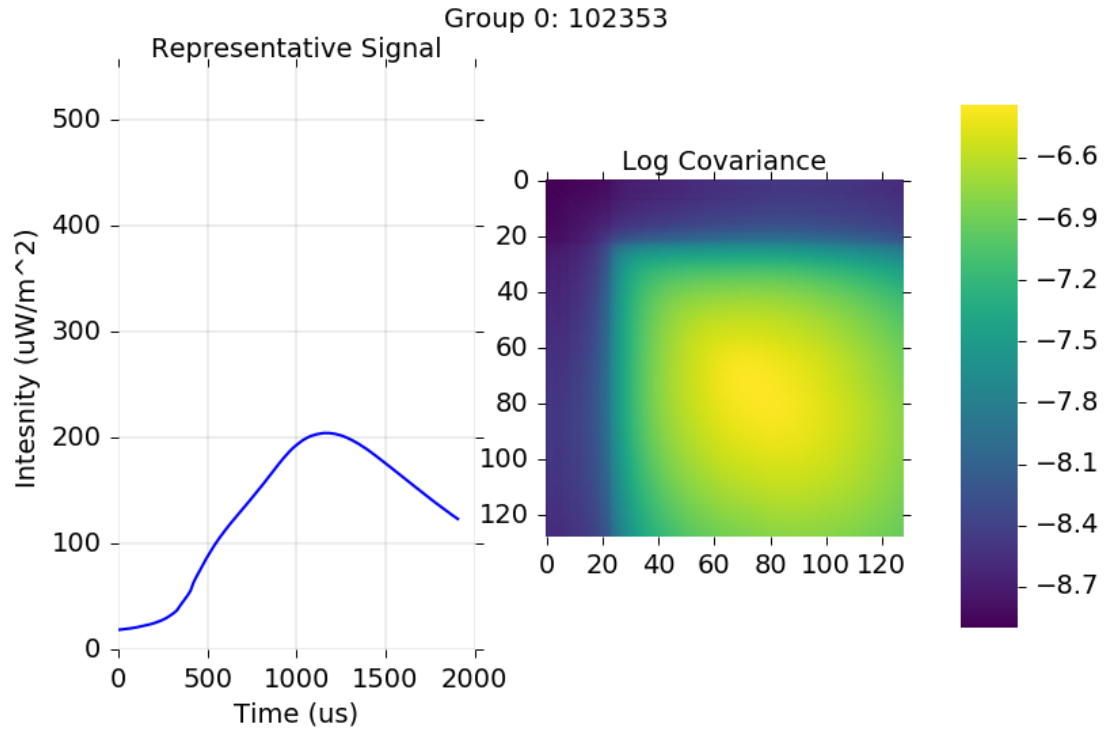

Figure 2.16: Group 0 representative signal and log covariance matrix.

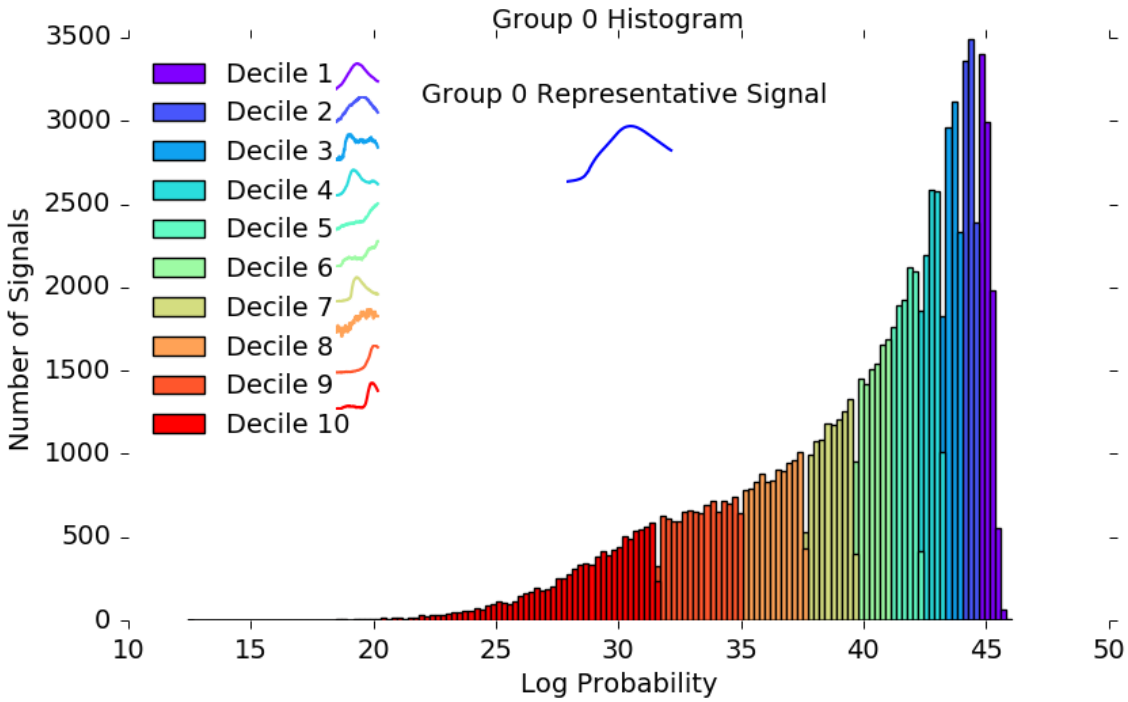

Figure 2.17: Group 0 decile breakdown. The lower the decile number in the legend (i.e. Decile 1), the higher the loglikelihood that a signal in this decile belongs in Group 0, and the more like the representative signal it is.

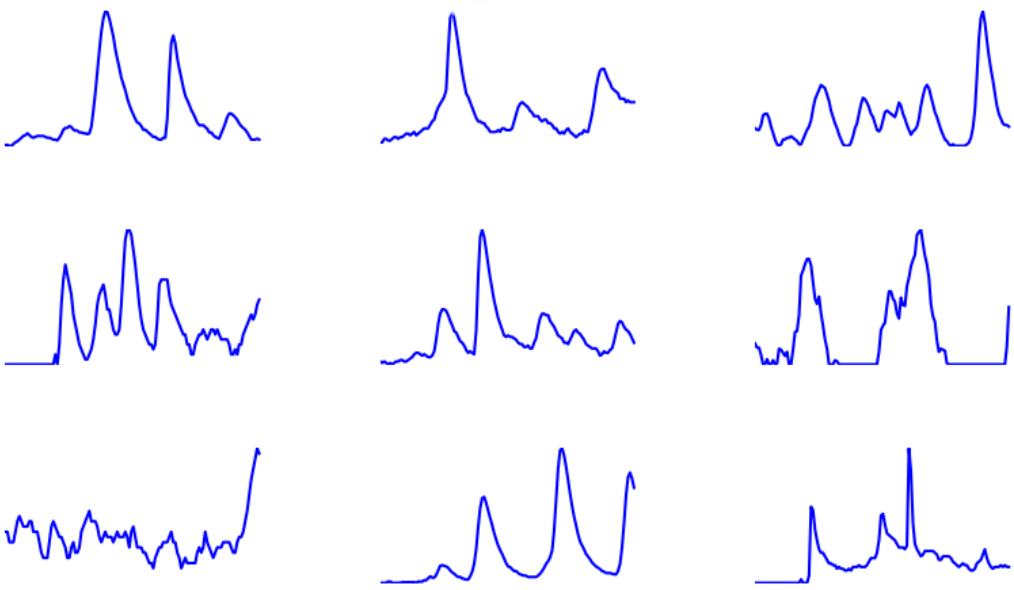

Figure 2.18: Examples from the anomalous set. The signals in this set have the lowest loglikelihoods (across all groups), indicating that they are the outliers of the lightning data.

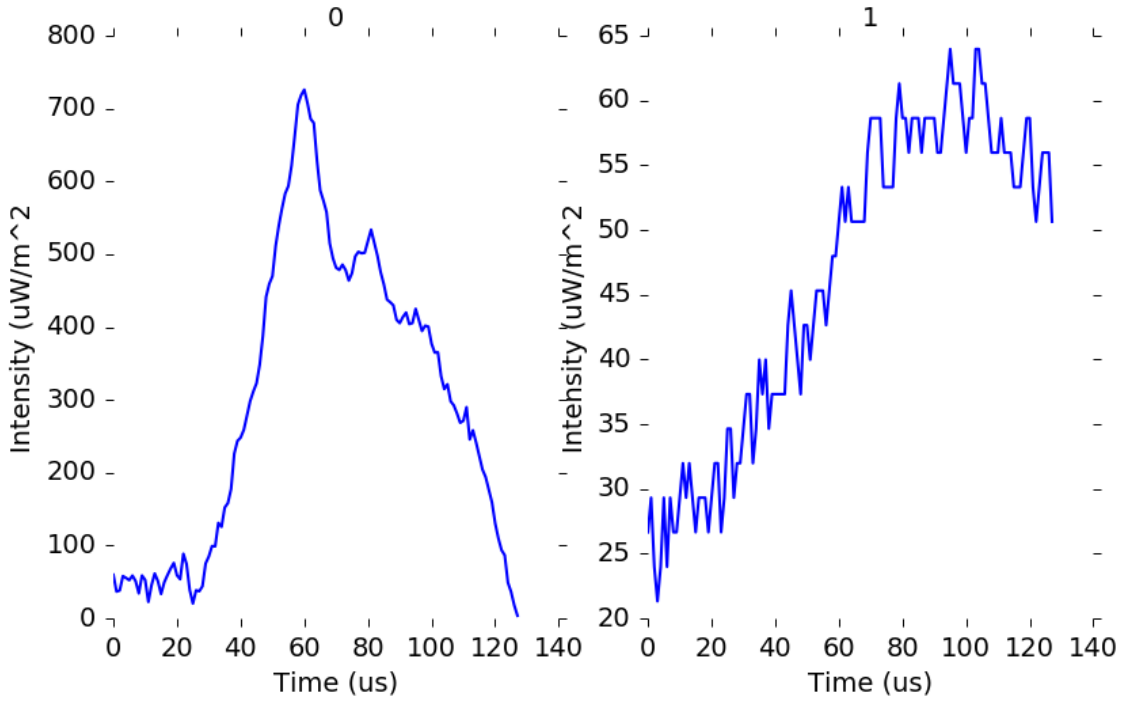

Figure 2.19: By treating lightning as a random realization of a Gaussian random vector with means and covariance matrices determined via familial clustering, we can synthesize new lightning signals for each class of signal. The right panel shows a measured lightning signal from Group 0; the left panel shows a synthesized lightning signal.

2.5 Conclusion and Future Work

Of the $914\,894$ signals in the original FORTÉ PDD 2001 data set, we found $38\,269$ to be non-lightning in the first stage (Section 2.3) and an additional $33\,827$ to be non-lightning in the second (Section 2.4.2). These $72\,096$ profiles corresponds to 7.9% of the total set. We then divided the remaining $842\,798$ signals into nine different families of lightning with representative means and covariance matrices for use in statistical models and lightning synthesis. We created a subset of $18\,000$ signals that efficiently cover the space of lightning profiles.

There is room for improvement in the area of familial clustering. Future work could explore alternative feature sets as well as subdividing groups into subgroups. We also note that many of the NNMF basis functions appear to be time-shifted versions of each other; directly parameterizing such time-shifts might drastically reduce the needed parameter space and add robustness at the cost of much more complicated machine learning procedures. One could then explore how much of the variation in the data comes from inherently different shapes, as opposed to time shifts resulting from similar lightning profiles occurring at difference distances that currently must be incorporated via a large set of basis functions. Such variation in time location could potentially be "normalized" in some fashion, analogous to the way we normalized amplitude variation via the L_2 norm.

Future work could further address more detailed characterizations of how well a set of representative signals cover the entire lightning signature space. The remainder of this thesis uses the exemplar from Group 5 as our canonical lightning template.

CHAPTER 3

ALGORITHMS FOR UNSATURATED OPTICAL MODELS

3.1 Introduction

This chapter discusses the maximum-likelihood estimation of TOA and nuisance parameters using uncensored Poisson and Gaussian probability models. The received OTS is modeled as a random realization of a discrete-time PPP with a time-varying intensity function, defined in (1.2). Although we test the algorithms in this chapter on data that has undergone saturation at a known level, the algorithms themselves are ignorant of the saturation effect. Awareness of saturation is introduced into the algorithms in Chapter 4. The unknown parameters are the TOA, τ, and the scaling factor, α. Hence, we derive the CRLB for each of these parameters. This chapter concludes with the results of several MC simulations.

3.2 Poisson Models

This section defines the uncensored Poisson loglikelihood, derives an algorithm for estimating the scaling factor, and presents associated CRLBs.

3.2.1 ML for Poisson Models

The likelihood of the uncensored Poisson distribution as applied to our model is

$$\mathcal{L}(\theta|x) = \prod_{i=1}^{N} \frac{\lambda_i^{x_i} e^{-\lambda_i}}{x_i!}, \tag{3.1}$$

where λ is defined in (1.2) and is a function of both α and τ, x is the received OTS, and $\theta = [\tau \;\; \alpha]^T$. Although x may have undergone saturation in practice, we do not include this fact in the algorithms in this chapter.

The loglikelihood of our Poisson model (3.1) is

$$\ell(\theta|x) = \sum_{i=1}^{N} \left(x_i \ln(\lambda_i) - \lambda_i - \ln(x_i!) \right). \tag{3.2}$$

Maximum-likelihood estimation of both α and τ requires a two-dimensional optimization. We can simplify the loglikelihood by ignoring all terms that do not depend on α or τ:

$$\begin{aligned} \ell(\theta|x) &= \sum_{i=1}^{N} x_i \ln(\lambda_i) - \sum_{i=1}^{N} \lambda_i \\ &= \sum_{i=1}^{N} x_i \ln(\alpha \mu_i^{(\tau)} + \lambda_0) - \alpha \sum_{i=1}^{N} \mu_i^{(\tau)}. \end{aligned} \tag{3.3}$$

If α is known, then $\hat{\tau}$, the ML estimate of the TOA, is given by:

$$\begin{aligned} \hat{\tau} &= \arg\max_{\tau} \ \ell(\theta|x) \\ &= \arg\max_{\tau} \ \sum_{i=1}^{N} x_i \ln(\alpha \mu_i^{(\tau)} + \lambda_0) - \alpha \sum_{i=1}^{N} \mu_i^{(\tau)}. \end{aligned} \tag{3.4}$$

This maximization generally requires some sort of brute-force search. If α is also unknown, it must be co-estimated. In such cases, our approach to finding $\hat{\tau}$ is to estimate α given a particular τ for each value of τ examined by the search algorithm. This is convenient because elegant iterative algorithms are available for estimating α given a particular τ.

3.2.2 α Estimation with Poisson Models

Since closed-form maximum-likelihood estimates of α are unavailable, iterative approaches must be explored. We are drawn to the conceptual elegance of Expectation-Maximization (EM) algorithms [19]. EM algorithms are constructed using mappings from idealized "complete" to the available "incomplete" data and are naturally applicable to many problems that can be framed in this mindset. EM algorithms operate by iteratively maximizing the likelihood of the complete data, while incorporating a conditional expectation to represent

hypothetical data that has not been actually observed.

An EM algorithm for the estimation of α has been derived and modified in [14, 20] for the case in which a PPP is defined by an intensity function that includes a scale factor times a signature plus an additive background counts component, as in (1.2). Our signal is modeled as the sum of a signal component with time-varying mean $\alpha\mu_i$, plus a background noise component with constant mean λ_0. Therefore, we can model the received signal as

$$x_i = x_i^S + x_i^B, \tag{3.5}$$

where x_i^S is a realization of a PPP with $\lambda_i^S = \alpha\mu_i$ and x_i^B is a realization of a PPP with $\lambda_i^B = \lambda_0$. The incomplete data is x, the observed sum of the two random variables, and the complete data involves knowing both x_i^S and x_i^B. This complete data decomposition (3.5) is analogous to Equation A1 in [14]. Using this decomposition, as well as the work outlined in [20], we uncover the following EM algorithm to estimate the scaling factor:

$$\alpha^{(j+1)} = \frac{1}{\sum_{i=1}^N \mu_i^{(\tau)}} \left(\sum_{i=1}^N \frac{x_i \mu_i^{(\tau)} \alpha^{(j)}}{\alpha^{(j)} \mu_i^{(\tau)} + \lambda_0} \right). \tag{3.6}$$

The algorithm terminates when $\alpha^{(j+1)}$ and $\alpha^{(j)}$ become sufficiently close.

As we seek to maximize over the search space for τ in (3.4), we solve (3.6) before evaluating the loglikelihood for any given τ.

3.2.3 CRLBs for Poisson Models

The CRLB is a theoretical lower bound on the variance of an unbiased estimator. The calculation of the bound entails computing the inverse of the Fisher information matrix (FIM). The FIM is defined in the following manner:

$$I(\theta) = \begin{bmatrix} -\mathbb{E}\left[\frac{\partial^2 \ell(\theta|x)}{\partial \tau^2}\right] & -\mathbb{E}\left[\frac{\partial^2 \ell(\theta|x)}{\partial \tau \partial \alpha}\right] \\ -\mathbb{E}\left[\frac{\partial^2 \ell(\theta|x)}{\partial \tau \partial \alpha}\right] & -\mathbb{E}\left[\frac{\partial^2 \ell(\theta|x)}{\partial \alpha^2}\right] \end{bmatrix}, \tag{3.7}$$

where $\theta = [\tau \;\; \alpha]^T$. In the case of the Poisson distribution, the FIM becomes

$$I(\theta) = \begin{bmatrix} \alpha^2 \sum_{i=1}^{N} \frac{\left(\partial \mu_i^{(\tau)} / \partial \tau\right)^2}{\alpha \mu_i^{(\tau)} + \lambda_0} & \alpha \sum_{i=1}^{N} \frac{\mu_i^{(\tau)} \left(\partial \mu_i^{(\tau)} / \partial \tau\right)}{\alpha \mu_i^{(\tau)} + \lambda_0} \\ \alpha \sum_{i=1}^{N} \frac{\mu_i^{(\tau)} \left(\partial \mu_i^{(\tau)} / \partial \tau\right)}{\alpha \mu_i^{(\tau)} + \lambda_0} & \sum_{i=1}^{N} \frac{\left(\mu_i^{(\tau)}\right)^2}{\alpha \mu_i^{(\tau)} + \lambda_0} \end{bmatrix}. \tag{3.8}$$

Inverting this matrix leads to the lower bounds

$$\text{var}(\tau) \geq \frac{\sum_{i=1}^{N} \frac{\left(\mu_i^{(\tau)}\right)^2}{\alpha \mu_i^{(\tau)} + \lambda_0}}{\alpha^2 \left(\sum_{i=1}^{N} \frac{\left(\partial \mu_i^{(\tau)} / \partial \tau\right)^2}{\alpha \mu_i^{(\tau)} + \lambda_0} \right) \left(\sum_{i=1}^{N} \frac{\left(\mu_i^{(\tau)}\right)^2}{\alpha \mu_i^{(\tau)} + \lambda_0} \right) - \alpha^2 \left(\sum_{i=1}^{N} \frac{\mu_i^{(\tau)} \left(\partial \mu_i^{(\tau)} / \partial \tau\right)}{\alpha \mu_i^{(\tau)} + \lambda_0} \right)^2} \tag{3.9}$$

$$\text{var}(\alpha) \geq \frac{\alpha^2 \sum_{i=1}^{N} \frac{\left(\partial \mu_i^{(\tau)} / \partial \tau\right)^2}{\alpha \mu_i^{(\tau)} + \lambda_0}}{\alpha^2 \left(\sum_{i=1}^{N} \frac{\left(\partial \mu_i^{(\tau)} / \partial \tau\right)^2}{\alpha \mu_i^{(\tau)} + \lambda_0} \right) \left(\sum_{i=1}^{N} \frac{\left(\mu_i^{(\tau)}\right)^2}{\alpha \mu_i^{(\tau)} + \lambda_0} \right) - \alpha^2 \left(\sum_{i=1}^{N} \frac{\mu_i^{(\tau)} \left(\partial \mu_i^{(\tau)} / \partial \tau\right)}{\alpha \mu_i^{(\tau)} + \lambda_0} \right)^2} \tag{3.10}$$

These bounds incorporate the fact that both τ and α are unknown; the lower bound on τ would be different if α were known and we were simply looking for τ.

3.3 Gaussian Models

This section defines the uncensored Gaussian loglikelihood, derives an algorithm for estimating the scaling factor, and presents associated CRLBs.

3.3.1 ML for Gaussian Models

The Gaussian distribution is a reasonable approximation for the Poisson distribution as the number of counts increases. The Gaussian distribution in the model described here has a mean and variance of λ, the intensity function describing the original PPP. The likelihood

function for the uncensored Gaussian distribution adapted to our model is

$$\mathcal{L}(\theta|x) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\lambda_i}} \exp\left\{-\frac{(x_i-\lambda_i)^2}{2\lambda_i}\right\}. \tag{3.11}$$

The associated loglikelihood is:

$$\ell(\theta|x) = -\frac{N}{2}\ln(2\pi) - \sum_{i=1}^{N}\frac{1}{2}\ln(\lambda_i) - \sum_{i=1}^{N}\frac{(x_i-\lambda_i)^2}{2\lambda_i}. \tag{3.12}$$

Ignoring all terms that do not depend on τ or α, (3.12) reduces to

$$\begin{aligned}\ell(\theta|x) &= -\sum_{i=1}^{N}\frac{1}{2}\ln(\lambda_i) - \sum_{i=1}^{N}\frac{(x_i-\lambda_i)^2}{2\lambda_i} \\ &= -\sum_{i=1}^{N}\frac{1}{2}\ln\left(\alpha\mu_i^{(\tau)}+\lambda_0\right) - \sum_{i=1}^{N}\frac{\left[x_i-\left(\alpha\mu_i^{(\tau)}+\lambda_0\right)\right]^2}{2\left(\alpha\mu_i^{(\tau)}+\lambda_0\right)}.\end{aligned} \tag{3.13}$$

If α is known, we estimate the TOA by solving the following optimization program:

$$\hat{\tau} = \arg\max_{\tau} \; -\frac{1}{2}\sum_{i=1}^{N}\ln\left(\alpha\mu_i^{(\tau)}+\lambda_0\right) - \frac{1}{2}\sum_{i=1}^{N}\frac{\left[x_i-\left(\alpha\mu_i^{(\tau)}+\lambda_0\right)\right]^2}{\alpha\mu_i^{(\tau)}+\lambda_0}. \tag{3.14}$$

As in the Poisson case described above, if α and τ are both unknown, our strategy is to estimate α for a given tau with an appealing "inner loop" iteration while a more brute-force "outer loop" searches over τ.

3.3.2 α Estimation with Gaussian Models

To estimate α for a given τ, we developed an EM algorithm that is similar in spirit to the Poisson algorithm in (3.6). To our knowledge, this thesis is the first presentation of this particular algorithm for Gaussian data. Hence, this section requires more detail than the review in Section 3.2.2, so we will need additional caution in clarifying the difference between random quantities (represented with capital letters) and realizations of those quan-

tities (represented with lower case letters).

Decomposing the OTS into a random signal component X_i^S, $i = 1, \cdots, N$, (a Gaussian random vector with mean and variance $\alpha\mu_i$) and a background component X_i^B, $i = 1, \cdots, N$ (a Gaussian random vector with mean and variance λ_0), the complete-data loglikelihood for α estimation becomes

$$\ell^C(\alpha|X^S, X^B) = -\frac{N}{2}\ln(\alpha) - \frac{1}{2}\alpha\sum_{i=1}^{N}\mu_i^{(\tau)} - \frac{1}{2\alpha}\sum_{i\in\mathcal{N}}\frac{(X_i^S)^2}{\mu_i^{(\tau)}}, \tag{3.15}$$

where terms lacking α have been left absent in anticipation of the maximization step.

The expectation step of our EM algorithm for iteration $j+1$ involves taking the expectation of (3.15) conditioned on the received data x and $\alpha^{(j)}$, the estimate of α from the previous iteration. This conditional expectation is commonly notated as $\mathcal{Q}(\alpha|\alpha^{(j)})$, where terms not containing α are disregarded in anticipation of the maximization step. The α to the left of the vertical bar corresponds to the α naturally appearing in (3.15). The $\alpha^{(j)}$ to the right of the vertical bar appears in the expressions arising from taking conditional expectations. For this complete-data loglikelihood, $\mathcal{Q}$ is

$$\mathcal{Q}(\alpha|\alpha^{(j)}) = -\frac{N}{2}\ln(\alpha) - \frac{1}{2}\alpha\sum_{i=1}^{N}\mu_i^{(\tau)} - \frac{C}{2\alpha}, \tag{3.16}$$

where

$$C = \sum_{i=1}^{N}\frac{\mathbb{E}\left[(X_i^S)^2|x_i;\alpha^{(j)}\right]}{\mu_i^{(\tau)}}. \tag{3.17}$$

The x_i instance to the right of the vertical bars in the expectation is shorthand for the event $X_i = x_i$.

Maximizing (3.16) over α for the M step of our EM algorithm is easily performed by taking the derivative of (3.16) with respect to α and setting equal to zero, resulting in the iteration

$$\alpha^{(j+1)} = \frac{-N + \sqrt{N^2 + 4C\sum_{i=1}^{N}\mu_i^{(\tau)}}}{2\sum_{i=1}^{N}\mu_i^{(\tau)}}. \tag{3.18}$$

The algorithm terminates when $\alpha^{(j+1)}$ and $\alpha^{(j)}$ become sufficiently close.

We must evaluate $\mathbb{E}\left[(X_i^S)^2|x_i;\alpha^{(j)})\right]$. Since X^S and X are jointly Gaussian, the conditional probability density function (PDF) of X_i^S conditioned on the event $X_i = x_i$ is

$$f_{X_i^S}(x_i^S|x_i;\alpha^{(j)}) = \frac{f_{X_i^S,X_i}(x_i^S,x_i;\alpha^{(j)})}{f_{X_i}(x_i;\alpha^{(j)})} = \frac{\exp\left\{\frac{-\left[x_i^S-\frac{\alpha^{(j)}\mu_i^{(\tau)}x_i}{\alpha^{(j)}\mu_i^{(\tau)}+\lambda_0}\right]^2}{2\alpha^{(j)}\mu_i^{(\tau)}\left(1-\frac{\alpha^{(j)}\mu_i^{(\tau)}}{\alpha^{(j)}\mu_i^{(\tau)}+\lambda_0}\right)}\right\}}{\sqrt{2\pi\alpha^{(j)}\mu_i^{(\tau)}\left(1-\frac{\alpha^{(j)}\mu_i^{(\tau)}}{\alpha^{(j)}\mu_i^{(\tau)}+\lambda_0}\right)}}, \tag{3.19}$$

and the conditional expectation is $\mathbb{E}\left[(X_i^S)^2|x_i;\alpha^{(j)})\right] =$

$$\alpha^{(j)}\mu_i - \frac{\left(\alpha^{(j)}\mu_i^{(\tau)}\right)^2}{\alpha^{(j)}\mu_i^{(\tau)}+\lambda_0} + \frac{\left(\alpha^{(j)}\mu_i^{(\tau)}x_i\right)^2}{\left(\alpha^{(j)}\mu_i^{(\tau)}+\lambda_0\right)^2}. \tag{3.20}$$

Plugging (3.20) into (3.18) yields the following EM iteration:

$$\alpha^{(j+1)} = -\frac{N}{2\sum_{i=1}^N \mu_i^{(\tau)}} + \frac{1}{2}\sqrt{\frac{N^2}{\left(\sum_{i=1}^N \mu_i^{(\tau)}\right)^2} + \frac{4\alpha^{(j)}}{\sum_{i=1}^N \mu_i^{(\tau)}}\Xi}, \tag{3.21}$$

where

$$\Xi = N - \alpha^{(j)}\sum_{i=1}^N \frac{\mu_i^{(\tau)}}{\alpha^{(j)}\mu_i^{(\tau)}+\lambda_0} + \alpha^{(j)}\sum_{i=1}^N \frac{\mu_i^{(\tau)}x_i^2}{\left(\alpha^{(j)}\mu_i^{(\tau)}+\lambda_0\right)^2}. \tag{3.22}$$

In the broader context of jointly estimating α and τ, a more brute-force approach is needed to find τ. Before evaluation of the loglikelihood of the Gaussian distribution for a given τ, we estimate the α using (3.21).

3.3.3 CRLBs for Gaussian Models

The FIM for the Gaussian case becomes $I(\theta) =$

$$\begin{bmatrix} \frac{\alpha^2}{2}\sum_{i=1}^{N}\frac{\left(\partial\mu_i^{(\tau)}/\partial\tau\right)^2}{\left(\alpha\mu_i^{(\tau)}+\lambda_0\right)^2}+\alpha^2\sum_{i=1}^{N}\frac{\left(\partial\mu_i^{(\tau)}/\partial\tau\right)^2}{\alpha\mu_i^{(\tau)}+\lambda_0} & \frac{\alpha}{2}\sum_{i=1}^{N}\frac{\mu_i^{(\tau)}\left(\partial\mu_i^{(\tau)}/\partial\tau\right)}{\left(\alpha\mu_i^{(\tau)}+\lambda_0\right)^2}+\alpha\sum_{i=1}^{N}\frac{\mu_i^{(\tau)}\left(\partial\mu_i^{(\tau)}/\partial\tau\right)}{\alpha\mu_i^{(\tau)}+\lambda_0} \\ \frac{\alpha}{2}\sum_{i=1}^{N}\frac{\mu_i^{(\tau)}\left(\partial\mu_i^{(\tau)}/\partial\tau\right)}{\left(\alpha\mu_i^{(\tau)}+\lambda_0\right)^2}+\alpha\sum_{i=1}^{N}\frac{\mu_i^{(\tau)}\left(\partial\mu_i^{(\tau)}/\partial\tau\right)}{\alpha\mu_i^{(\tau)}+\lambda_0} & \frac{1}{2}\sum_{i=1}^{N}\frac{\left(\mu_i^{(\tau)}\right)^2}{\left(\alpha\mu_i^{(\tau)}+\lambda_0\right)^2}+\sum_{i=1}^{N}\frac{\left(\mu_i^{(\tau)}\right)^2}{\alpha\mu_i^{(\tau)}+\lambda_0} \end{bmatrix} \tag{3.23}$$

Inverting this matrix leads to the lower bounds

$$\text{var}(\tau) \geq \frac{\frac{1}{2}\sum_{i=1}^{N}\frac{\left(\mu_i^{(\tau)}\right)^2}{\left(\alpha\mu_i^{(\tau)}+\lambda_0\right)^2}+\sum_{i=1}^{N}\frac{\left(\mu_i^{(\tau)}\right)^2}{\alpha\mu_i^{(\tau)}+\lambda_0}}{\beta} \tag{3.24}$$

$$\text{var}(\alpha) \geq \frac{\frac{\alpha^2}{2}\sum_{i=1}^{N}\frac{\left(\partial\mu_i^{(\tau)}/\partial\tau\right)^2}{\left(\alpha\mu_i^{(\tau)}+\lambda_0\right)^2}+\alpha^2\sum_{i=1}^{N}\frac{\left(\partial\mu_i^{(\tau)}/\partial\tau\right)^2}{\alpha\mu_i^{(\tau)}+\lambda_0}}{\beta}, \tag{3.25}$$

where $\beta =$

$$\left(\frac{\alpha^2}{2}\sum_{i=1}^{N}\frac{\left(\partial\mu_i^{(\tau)}\Big/\partial\tau\right)^2}{\left(\alpha\mu_i^{(\tau)}+\lambda_0\right)^2}+\alpha^2\sum_{i=1}^{N}\frac{\left(\partial\mu_i^{(\tau)}\Big/\partial\tau\right)^2}{\alpha\mu_i^{(\tau)}+\lambda_0}\right)\left(\frac{1}{2}\sum_{i=1}^{N}\frac{\left(\mu_i^{(\tau)}\right)^2}{\left(\alpha\mu_i^{(\tau)}+\lambda_0\right)^2}+\sum_{i=1}^{N}\frac{\left(\mu_i^{(\tau)}\right)^2}{\alpha\mu_i^{(\tau)}+\lambda_0}\right)$$
$$-\left(\frac{\alpha}{2}\sum_{i=1}^{N}\frac{\mu_i^{(\tau)}\left(\partial\mu_i^{(\tau)}\Big/\partial\tau\right)}{\left(\alpha\mu_i^{(\tau)}+\lambda_0\right)^2}+\alpha\sum_{i=1}^{N}\frac{\mu_i^{(\tau)}\left(\partial\mu_i^{(\tau)}\Big/\partial\tau\right)}{\alpha\mu_i^{(\tau)}+\lambda_0}\right)^2.$$

3.4 Ad Hoc Approaches

The purpose of the ad hoc methods outlined below is to inform the TOA estimation algorithms described above about saturation in the simplest way possible. The optimization programs of (3.4) and (3.14) remain unchanged; for each candidate τ, the algorithms described above are tried on saturated data to compute α. Since those EM algorithms are unaware of the saturation, we would expect a decline in performance in estimating α.

After estimating α, λ_i is calculated according to (1.2) and then thresholded at the known saturated level. This saturated λ_i is used instead of the original λ_i in the optimization problem in (3.4) or (3.14). In this way, the procedure takes into account saturation at a very rudimentary level, without complicating the algorithms.

3.5 Results

This section shows the results of several Monte Carlo (MC) simulations. These simulations include an analysis of the effect of needing to co-estimate α as opposed to knowing α *a priori*, a study of the effectiveness of the suggested ad hoc methods, and a comparison of the Poisson and Gaussian models. We begin by outlining the setup for our MC simulations and present our method for solving the one-dimensional optimization programs over possible τ described in (3.4) and (3.14).

3.5.1 Monte Carlo Setup

This subsection outlines the setup for all MC simulations in this thesis. For each MC simulation, we start with the lightning signature shown in Fig. 1.1. Since this lightning signature is 128 time samples long with a sample period of 15 μsec between samples, we designate a receive window of 5.76 msec consisting of 384 time samples. This allows enough space for three full lightning signatures in the receive window. We randomly assign one value for τ for all iterations of the MC simulation, and time shift the lightning signature to the proper arrival time inside the receive window. Note that τ is in units of μsec. At this point, the receive window represents $\mu^{(\tau)}$, the time-shifted template of the lightning.

For each MC simulation, we assign a static value for both T and λ_0, and sweep the value of α across a range. For each value of α, we calculate λ according to (1.2). For each MC run, a random realization of a PPP with the intensity λ is generated and then saturated at the threshold T. We then run our algorithms to estimate α and τ given the received signal.

To find the ML of $\hat{\tau}$ as previously described, one option is to time-shift the intensity

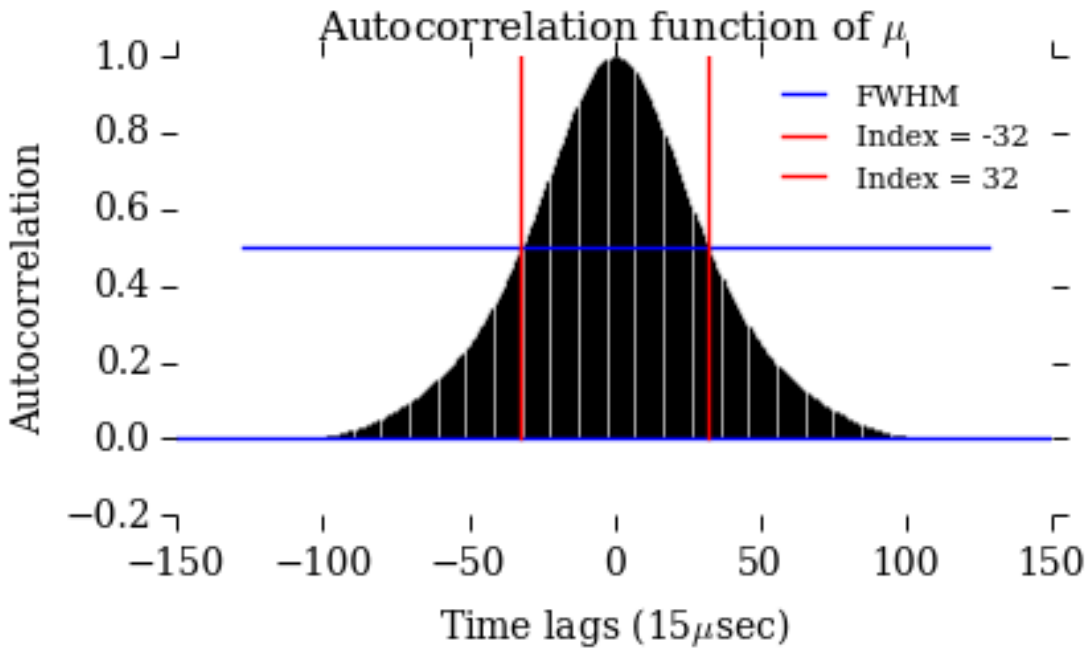

Figure 3.1: Full width at half maximum of the lightning signature, which is used as the resolution of the first sweep across τ for likelihood maximization.

template for every possible value of τ in the receive window, estimate α at that location, compute the loglikelihood, and select the value of τ that results in the largest loglikelihood. Such an exhaustive search is computationally intensive, so we opted for a multiresolution approach based on the "full width at half maximum" of the autocorrelation (shown in Fig. 3.1) of the template in Fig. 1.1. Analysis of Fig. 3.1) shows that the width at half maximum is 64 time samples, i.e. 960 µsec, long.

For each MC iteration, we sweep through the possible range of τ twice. The first sweep evaluates the loglikelihood at τ values spaced 32 time samples, i.e. 480 µsec, apart. Although Fig. 3.1 suggests we could afford a resolution up to 960 µsec for the first sweep, we decided to use half of this to avoid the scenario where the received signal falls midway between two adjacent resolution windows. We select the best estimate of τ from the first sweep and choose this as the center of a finer search on the second sweep. This second sweep evaluates the loglikelihood for τ values surrounding the center of the first sweep.

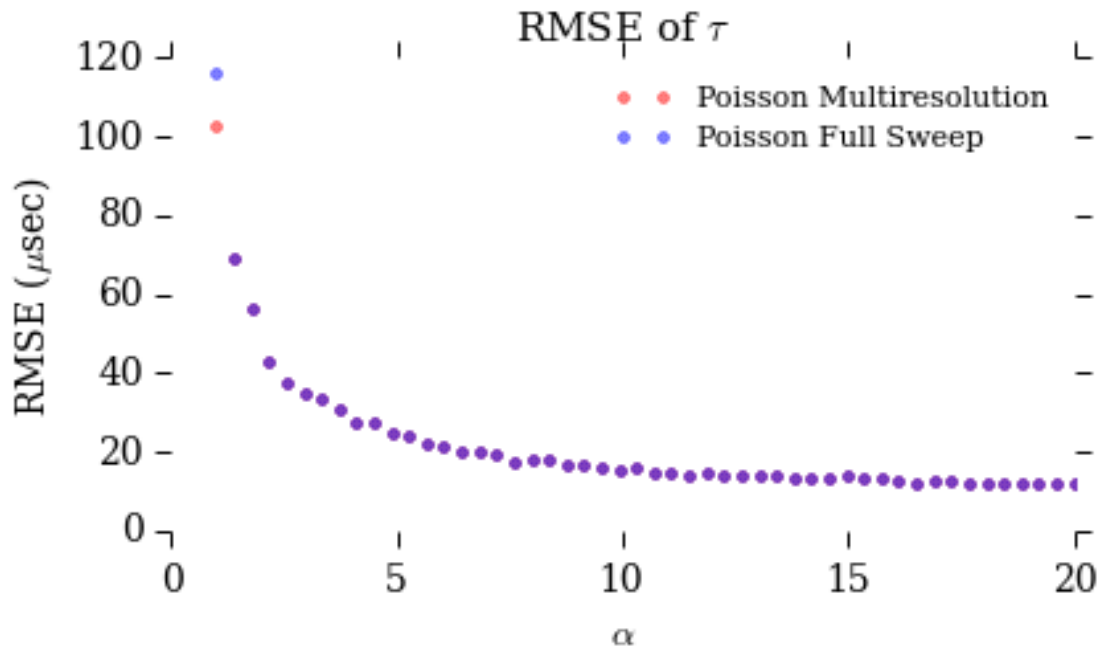

Figure 3.2: Performance comparison between full sweep and multiresolution approaches. Using a multiresolution approach to estimate the TOA allows for reduced computational complexity without loss of performance.

To test our assumption that this multiresolution scheme results in results comparable to computing the loglikelihood for all possible values of τ, we ran a MC simulation. For each run, we computed an estimate for τ using both methods, sweeping every value of τ (denoted as Poisson Full Sweep), as well as using the multiresolution scheme outlined above (denoted as Poisson Multiresolution). Figure 3.2 shows this comparison for the Poisson distribution. The two methods result in identical mean squared error in all but the lowest of α values.

All the MC simulations throughout this rest of this thesis follow the basic setup described above.

3.5.2 MC Simulations

This subsection describes the MC simulations we ran to evaluate the performance of the estimators derived previously in this chapter. For all simulations, we swept α from 1 to 20

in 100 steps, performing 1000 MC runs for each value of α.

We ran three different simulations. The first used a threshold of $T = 10$ and a background count intensity of $\lambda_0 = 1$. We chose these parameters to represent a low noise case where, for lower values of α, no saturation occurs, but as α increases, saturation begins to impact the algorithms. The second simulation again has $T = 10$, but the background intensity increases to $\lambda_0 = 7$. This represents a lower signal-to-noise ratio scenario. The final simulation reduces the threshold to $T = 5$ and the background noise counts to $\lambda_0 = 3$. In this case, noise is again lower (especially compared to larger values of α), but saturation plays a significant role for almost every received signal. For each simulation, we recorded mean squared errors for both τ and α for all aforementioned estimation algorithms (Poisson, Poisson Ad Hoc, Gaussian, and Gaussian Ad Hoc) for both known and unknown α. To make the plots easier to read, we plot root mean squared errors and the square-root of the CRLBs.

The estimation algorithm performed better (i.e. with a smaller mean squared error) when it was forced to co-estimate α as opposed to knowing α a priori. At first glance, this seems surprising. Figure 3.3 shows two examples of this, one for the Poisson distribution when $T = 10$ and $\lambda_0 = 7$ (left), and one for the Gaussian ad hoc case when $T = 5$ and $\lambda_0 = 3$ (right). While the effect is significantly more pronounced in the Gaussian ad hoc case with a lower threshold, it is true for all experiments we have conducted. The saturation effect is the most likely culprit of this seemingly strange behavior. If α is locked ahead of time, then there is no part of the algorithm that can adapt for the effects of saturation. However, if we let the algorithm estimate α, it is able to adapt the estimation of α to partially incorporate saturation by effectively underestimating α.

Figure 3.4 compares the Poisson estimator against the Poisson ad hoc estimator for the scenario of $T = 10$, $\lambda_0 = 7$ for both the α known a priori and α co-estimated cases. The Poisson estimator outperforms its ad hoc counterpart in both scenarios. The difference is particularly notable in the case when α is known. Again, this is most likely due to

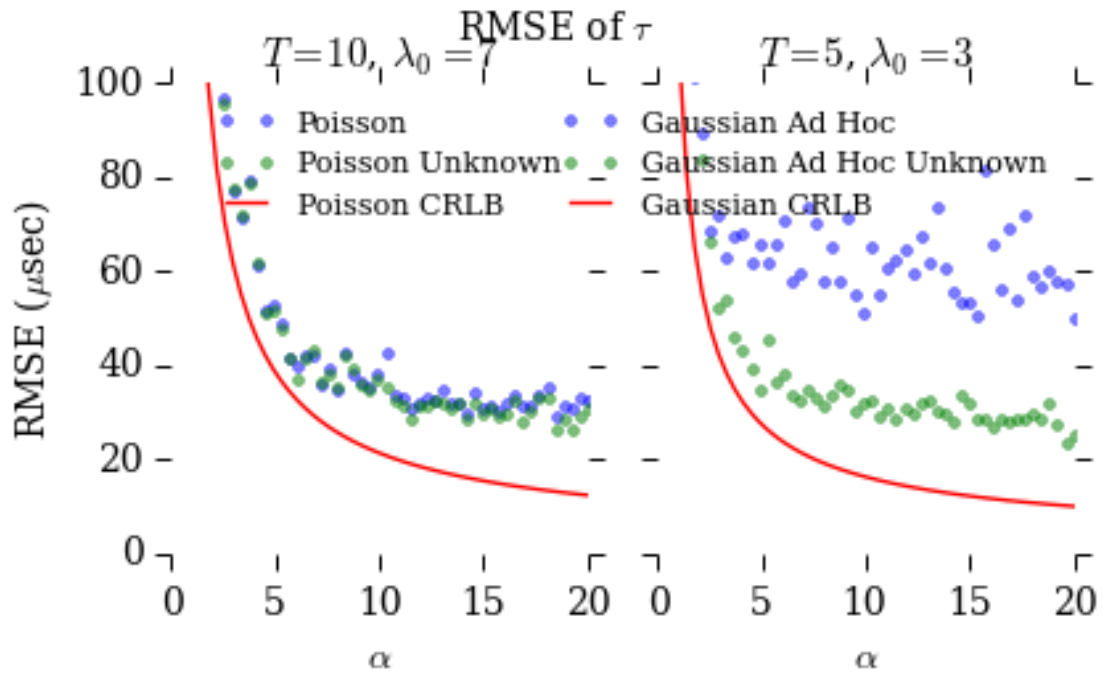

Figure 3.3: α co-estimation increases performance of estimation algorithms for uncensored estimators.

the effects of saturation. If α is known a priori, then saturating the template is a rigid operation that can not adapt to the received data the way it can if α is co-estimated. When α must be co-estimated, the performance gap is much smaller, although the Poisson estimator still maintains superior performance. The Poisson and Poisson ad hoc methods perform identically for α estimation, although this makes sense because the ad hoc method uses the same algorithm as the standard method for α estimation. This trend holds for all other scenarios tested for both the Poisson and Gaussian distributions.

Figures 3.5 - 3.7 illustrate the performance of the Poisson and Gaussian estimators, with α co-estimated, for all three scenarios. In all scenarios, the Poisson-based estimator performs better when estimating α. As the scenarios become less ideal (i.e. higher noise or lower threshold), the performance gap between the estimation algorithms increases, indicating that the Gaussian approximation does not work as well for estimating the scaling factor.

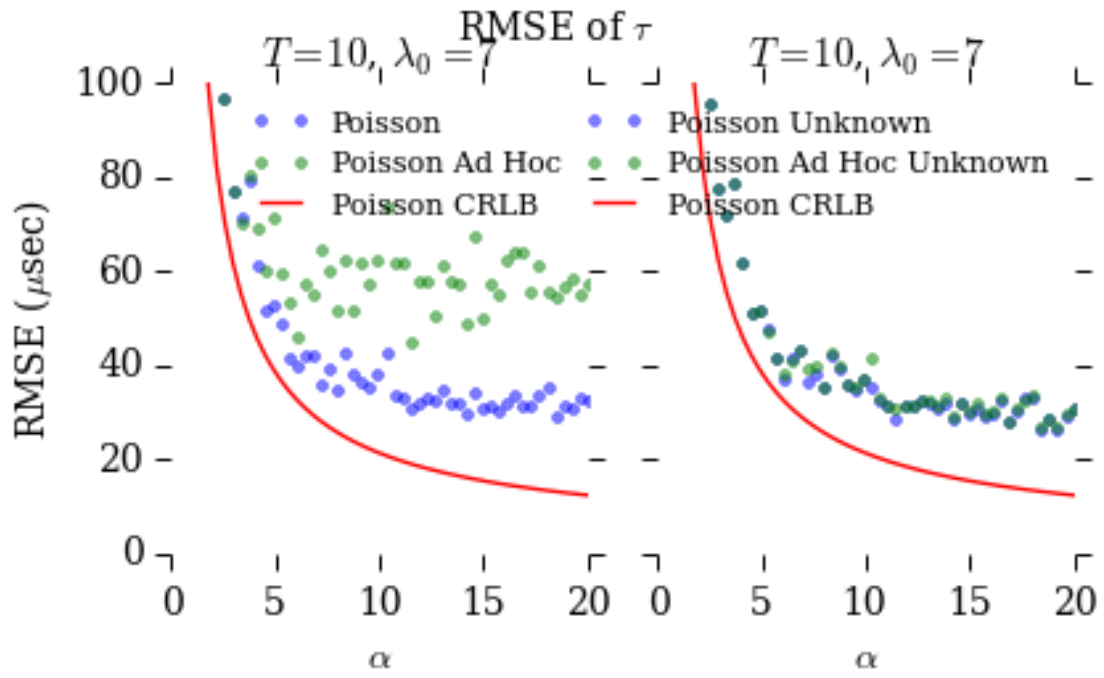

Figure 3.4: Traditional estimation algorithms outperform our ad hoc methods.

In the low noise scenario, the Poisson-based estimator performs better for TOA estimation than the Gaussian-based estimator, as might be expected. However, as the scenarios become less ideal, the Gaussian estimator begins to outperform the Poisson estimator. As previously noted, the Gaussian α estimation performs worse (in comparison with the Poisson model) as the scenario becomes less ideal. However, there appears to be an inverse relationship between α estimation and τ estimation with the Gaussian estimator: the worse the α estimation does in comparison to the Poisson α estimation, the better the Gaussian TOA estimation does in comparison to that of the Poisson. Hence, we suspect that the performance transition is directly related to estimating α. As with the previous scenarios analyzed, knowing the α value *a priori* actually hindered TOA estimation for all estimators, whereas when the scaling factor was estimated, it allowed for the saturation effect to be intrinsically – albeit unintentionally – incorporated. We conjecture that this observation can also help explain the performance curiosities described in the paragraph. As the noise increases and the threshold decreases, the Poisson model does a better job of estimating

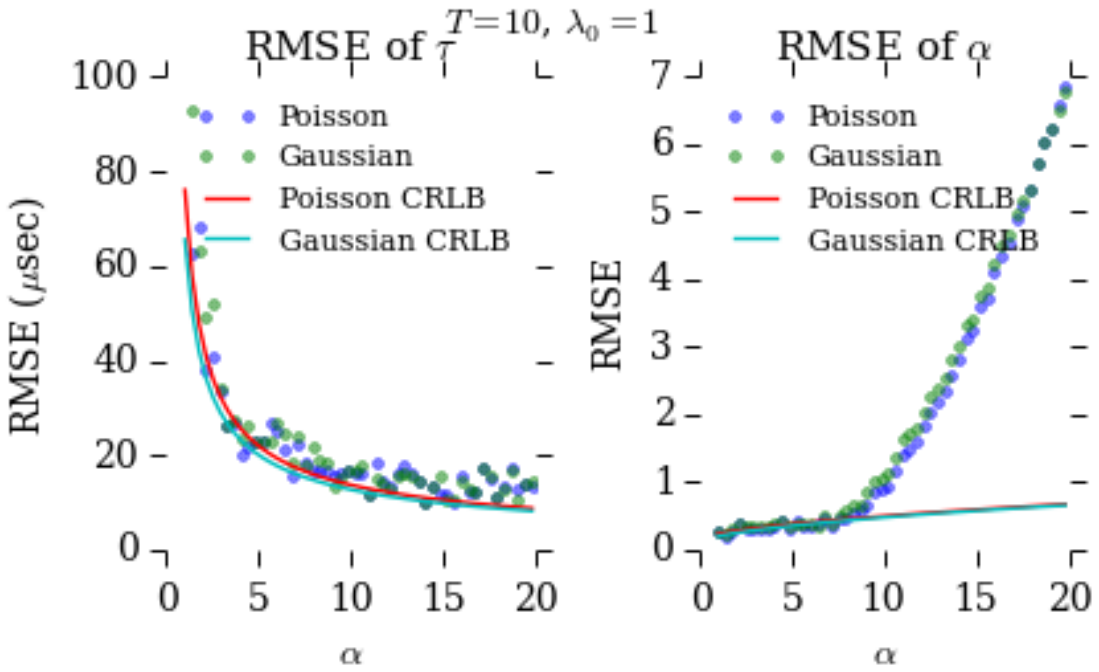

Figure 3.5: The Poisson estimator outperforms the Gaussian estimator in the low noise case.

the true value of α, but its resulting estimator does not adapt quite as much for the effect of saturation. The Gaussian distribution, on the other hand, does an inferior job of estimating the true α value, but it adapts better for saturation effects and results in better τ estimations. In any case, the models and estimation algorithms of Chapter 4, which directly incorporate the effects of saturation, beat all of the algorithms described in this chapter, so we see little benefit in further probing such apparent anomalies.

In all scenarios for all estimators, the lower bound outlined by the CRLBs is not achieved. For lower values of α, before the saturation takes effect, each estimator comes quite close to reaching this lower bound. However, as α increases and more of the received signal becomes saturated, the performance gap of the estimators, for both α and τ, between the CRLBs and the results of the simulations increases. This performance gap is especially obvious for α estimation, since the saturation effect more directly effects this process.

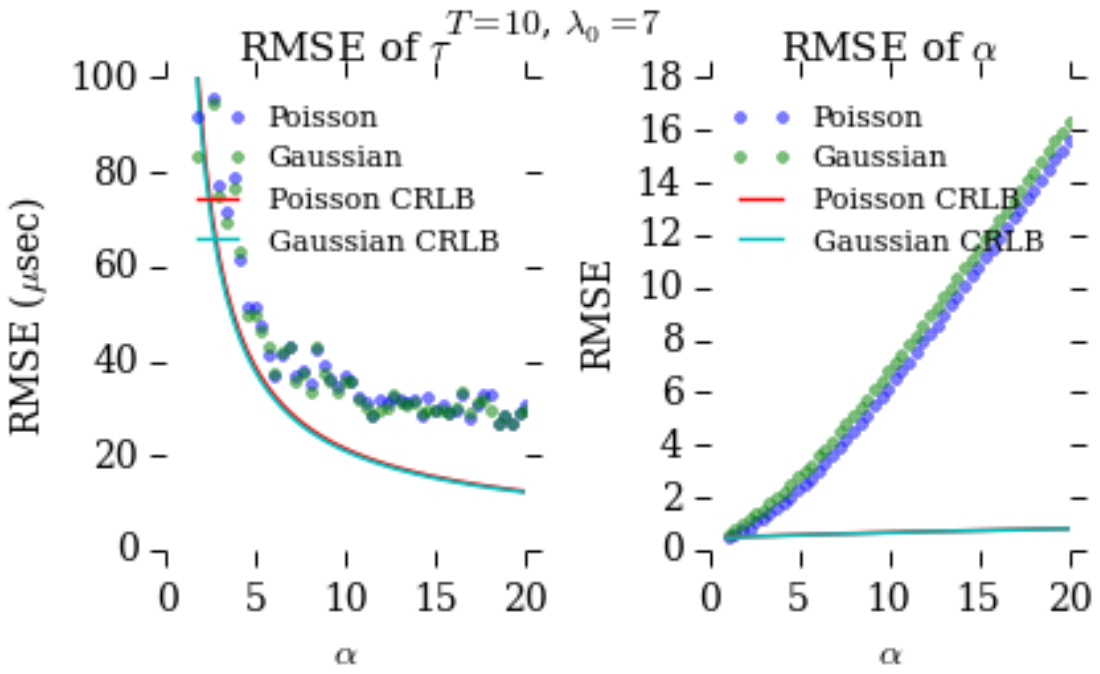

Figure 3.6: The Gaussian estimator outperforms the Poisson estimator in the high noise case.

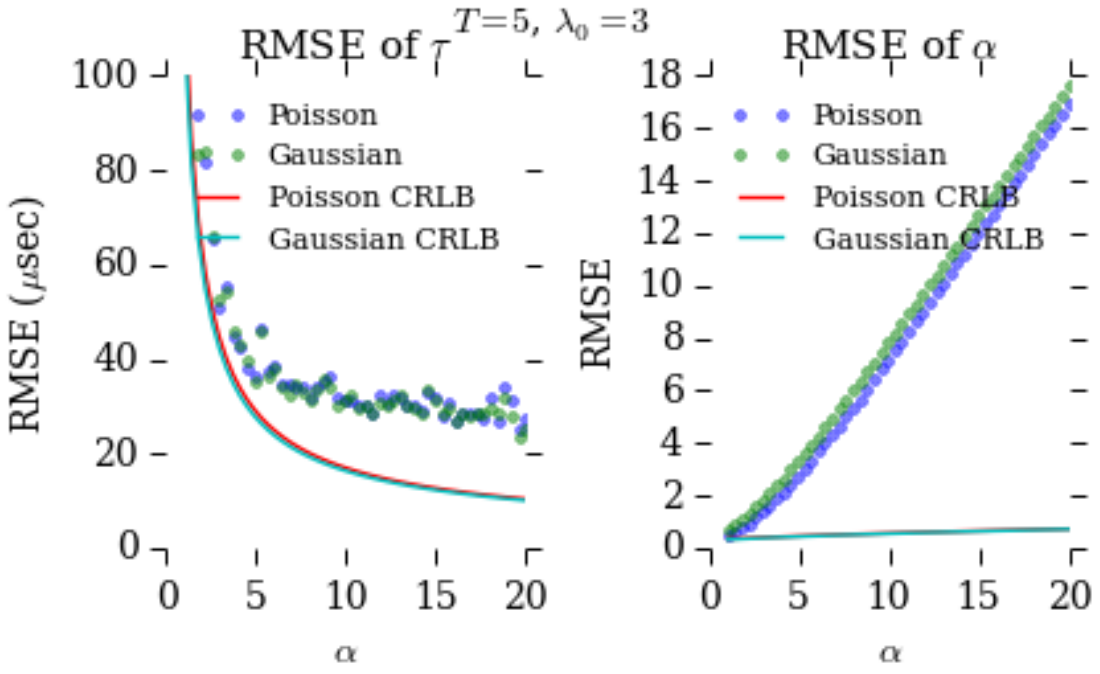

Figure 3.7: The Gaussian estimator outperforms the Poisson estimator in the low threshold case.

3.6 Conclusion

This chapter derived the ML procedures for TOA estimation, EM algorithms for scaling factor estimation, and CRLBs for τ and α for Poisson and Gaussian model. It also suggested an ad hoc way to incorporate the effects of saturation at a rudimentary level without significantly complicating our computations. Finally, this chapter described results of three MC simulations comparing these estimation algorithms in different scenarios.

As explained in Section 3.5, the best estimation performance was counterintuitively achieved when α was co-estimated, for both the Poisson and Gaussian models. The Poisson estimators performed better in low noise, ideal conditions, while the Gaussian estimators performed better in less than ideal conditions. Hence, both of these estimators will be compared to the censored estimators described in later chapters. While the ad hoc methods did not perform poorly, they did not outperform the aforementioned methods in any scenarios, and thus will not be considered further.

CHAPTER 4

ALGORITHMS FOR SATURATED OPTICAL MODELS

4.1 Introduction

Chapter 3 derived algorithms TOA and scaling factor estimation for OTSs using uncensored Poisson and Gaussian distributions. This chapter formulates analogous algorithms using censored Poisson and Gaussian distributions to directly address saturation effects.

Censored probability distributions exhibit a "binning" effect. In our censored probability models, if a received value is smaller than the saturation threshold of the sensor, the regular Poisson or Gaussian likelihood functions are. However, if the received value is equivalent to the threshold, indicating that saturation has occurred, then the likelihood consists of the discrete sum or continuous integral, as appropriate, of the uncensored Poisson or Gaussian distribution from the threshold to infinity. This represents the likelihood that the received value consisted of a "true" value at or above the threshold saturation threshold. For both censored Poisson and censored Gaussian models,

$$\mathcal{L}(\theta|x) = \prod_{i=1}^{N} a_i(t), \tag{4.1}$$

where

$$a_i(t) = \begin{cases} \frac{\exp\{-\lambda_i\}\lambda_i^{x_i}}{x_i!} & \text{for } x_i < T \\ 1 - \exp\{-\lambda_i\} \sum_{j=0}^{\lfloor T \rfloor} \frac{\lambda_i^j}{j!} & \text{for } x_i \geq T \end{cases} \tag{4.2}$$

for the censored Poisson model and

$$a_i(t) = \begin{cases} \frac{1}{\sqrt{2\pi\lambda_i}} \exp\left\{-\frac{(x_i-\lambda_i)^2}{2\lambda_i}\right\} & \text{for } x_i < T \\ 1 - \frac{1}{\sqrt{2\pi\lambda_i}} \int_{-\infty}^{T} \exp\left\{-\frac{(t-\lambda_i)^2}{2\lambda_i}\right\} dt & \text{for } x_i \geq T \end{cases} \tag{4.3}$$

for the censored Gaussian model.

The corresponding loglikelihoods are $\ell(\alpha|x) = \sum_{i=1}^{n} h_i(t)$, where

$$h_i(t) = \begin{cases} x_i \ln(\lambda_i) - \lambda_i - \ln(x_i!) & \text{for } x_i < T \\ \ln\left(1 - e^{-\lambda_i} \sum_{j=0}^{\lfloor T \rfloor} \frac{\lambda_i^j}{j!}\right) & \text{for } x_i \geq T \end{cases} \tag{4.4}$$

for the censored Poisson model and

$$h_i(t) = \begin{cases} -\frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln(\lambda_i) - \frac{(x_i-\lambda_i)^2}{2\lambda_i} & x_i < T \\ \ln\left(1 - \frac{1}{\sqrt{2\pi\lambda_i}} \int_{-\infty}^{T} e^{-\frac{(t-\lambda_i)^2}{2\lambda_i}} dt\right) & x_i \geq T \end{cases}. \tag{4.5}$$

for the censored Gaussian model. As with the uncensored models, we estimate the TOA and scaling factor by maximizing the appropriate likelihood over the parameters α and τ. If α is assumed known, then estimating τ just requires a one-dimensional search. Because elegant iterative algorithms for estimating α given a particular τ are available, we approach co-estimation from the standpoint of an "outer loop" that tries different τ values and an "inner loop" that estimates α for those different τ value.

The remainder of this chapter focuses on the formulation of two EM algorithms for estimating the scale parameter α in (1.2) for a received optical signal that has undergone saturation. One algorithm is derived using the censored Poisson likelihood function. Since the Poisson distribution is well approximated by the Gaussian distribution as the number of photon counts increases, this chapter also explores an EM algorithm derived using a censored Gaussian distribution with mean of λ and variance of λ. These algorithms rely on conditional expectations that include either a sum or integral across an infinite range with no obvious closed-form solution, but the sums and integrals can be readily restricted to reasonable ranges without undue approximation error.

EM algorithms appear to be quite popular in applications involving Poisson data and/or censored data. [21] derives both deterministic and Monte-Carlo based EM algorithms for censored Gaussian, Laplace, and Rayleigh distributions. As explained in more detail in

Chapter 3, the authors of [14] and [20] developed and adapt an EM algorithm for the estimation of α given a uncensored PPP modeled along the lines of (1.2), with a scaled template and additive background. Their algorithm does not include saturation effects. Nonetheless, their algorithm was implemented in Section 3.5 to demonstrate the importance of taking censoring into account. Likewise, the authors of [22] discuss an EM algorithm for censored Poisson data. However, they only address Type I censoring of pooled histogram data, rather than the time-domain data models employed here. The algorithm in [22] could be derived as a simplified special case of our algorithm in which the signal template μ is constant and the differing count values are pooled across time.

Most of the remainder of this chapter ignores τ. We assume that the template is properly time-aligned with the received OTS. As such, the previously defined nomenclature of $\mu_i^{(\tau)}$ to indicate the dependency on τ shall be simplified to μ_i to reflect this reduced scope. Furthermore, this chapter exclusively uses the censored Poisson and Gaussian distributions, rather than the uncensored distributions of Chapter 3. Unless explicitly said otherwise, mentions to Poisson or Gaussian distributions in this chapter refer to their censored versions.

Section 4.2 derives our EM algorithms for estimating α given the signal model of (1.2) using the Poisson likelihood. It also discuss the algorithm's implementation in detail, particularly approximations of needed conditional expectations. Section 4.3 derives the equivalent EM algorithm for Gaussian models and discusses similar implementation details. Experimental results are presented in Section 4.4, which compares our new EM algorithms with the simpler of algorithms of Chapter 3 that are unaware of the censoring, as well as to each other.

4.2 EM for Poisson Data

This section derives our EM algorithm for estimating the scaling factor of the OTS modeled as a PPP.

The loglikelihood for the actual collected data, referred to in EM nomenclature as the

"incomplete-data" loglikelihood, is:

$$\ell^I(\alpha|x) = \sum_{i=1}^{n} h_i(t), \text{ where } \lambda_i = \alpha\mu_i + \lambda_0 \text{ and}$$
$$h_i(t) = \begin{cases} x_i \ln(\lambda_i) - \lambda_i - \ln(x_i!) & \text{for } x_i < T, \\ \ln\left(1 - e^{-\lambda_i} \sum_{j=0}^{\lfloor T \rfloor} \frac{\lambda_i^j}{j!}\right) & \text{for } x_i \geq T, \end{cases} \tag{4.6}$$

where T is the saturation threshold. The "complete" data, in this scenario, is a little more complicated than simply assuming we know the values of the received waveform before saturation.

Assume, for the moment, that our received signal x is not saturated. As shown in (1.2), our signal is modeled as the sum of a signal component with time-varying mean $\alpha\mu_i$, plus a background noise component with constant mean λ_0.[1] In the uncensored case, the received waveform could be modeled as

$$x = x^S + x^B, \tag{4.7}$$

where x^S is a realization of a PPP with $\lambda_i^S = \alpha\mu_i$ and x^B is a realization of a PPP with $\lambda_i^B = \lambda_0$. In the absence of saturation, the incomplete data is x, the observed sum of the two random variables, and the complete data involves knowing both x^S and x^B. This is identical to the signal decomposition used to derive the EM algorithms of Chapter 3.

To reintroduce saturation, define the sets $\mathcal{C}$ and $\mathcal{N}$, which respectively represent the indices of the censored and uncensored measurements. Obviously, $x_i = T$ for $i \in \mathcal{C}$. Let the new sequence y_i, $i \in \mathcal{C}$, represent the hypothetical knowledge of the true values of the censored time samples. We must apply the signal-plus-background decomposition illustrated in (4.7) to y:

$$y_i = y_i^S + y_i^B, \text{ for } i \in \mathcal{C}, \tag{4.8}$$

where $Y_i^S \sim Poisson(\alpha\mu_i)$ and $Y_i^B \sim Poisson(\lambda_0)$.

[1] The algorithm could be easily extended to incorporate time-varying background noise.

Since we do not have direct access to our hypothetical complete data, we express the complete-data loglikelihood in terms of random variables:

$$\begin{aligned}
&\ell^C(\alpha|X^S, X^B, Y^S, Y^B) \\
&\quad = -\sum_{i=1}^{N} \alpha\mu_i - \sum_{i=1}^{N} \lambda_0 \\
&\quad + \sum_{i\in\mathcal{N}} X_i^S \ln(\alpha\mu_i) + \sum_{i\in\mathcal{N}} X_i^B \ln(\lambda_0) \\
&\quad + \sum_{i\in\mathcal{C}} Y_i^S \ln(\alpha\mu_i) + \sum_{i\in\mathcal{C}} Y_i^B \ln(\lambda_0).
\end{aligned} \tag{4.9}$$

The expectation step of our EM algorithm for iteration $j+1$ involves taking the expectation of (4.9) conditioned on the received data x and $\alpha^{(j)}$, the estimate of α from the previous iteration. This conditional expectation is commonly notated as $\mathcal{Q}(\alpha|\alpha^{(j)})$, where terms not containing α are dropped in anticipation of the maximization step. The α to the left of the vertical bar corresponds to the α naturally appearing in (4.9). The $\alpha^{(j)}$ to the right of the vertical bar appears in the expressions arising from taking conditional expectations. For this complete-data loglikelihood, $\mathcal{Q}$ is

$$\mathcal{Q}(\alpha|\alpha^{(j)}) = -\alpha \sum_{i=1}^{N} \mu_i + C\ln(\alpha), \tag{4.10}$$

where

$$C = \sum_{i\in\mathcal{N}} \mathbb{E}\left[X_i^S|x_i;\alpha^{(j)}\right] + \sum_{i\in\mathcal{C}} \mathbb{E}\left[Y_i^S|x_i;\alpha^{(j)}\right]. \tag{4.11}$$

The x_i instance to the right of the vertical bars in the expectations are shorthand for the event $X_i = x_i$. The maximization step of our EM algorithm is straightforward. Maximizing (4.10) over α is readily performed by taking the derivative of (4.10) with respect to α and setting it equal to zero, yielding the iteration

$$\alpha^{(j+1)} = \frac{C}{\sum_{i=1}^{N} \mu_i}. \tag{4.12}$$

The algorithm terminates when $\alpha^{(j+1)}$ and $\alpha^{(j)}$ become sufficiently close.

Unfortunately, the evaluation of C is less straightforward. Starting with the first expectation, $\mathbb{E}\left[X_i^S|x_i;\alpha^{(j)}\right]$, note the PMF of X_i^S conditioned on the event $X_i = x_i$ is a binomial distribution:

$$\begin{aligned} f_{X_i^S}(x_i^S|x_i;\alpha^{(j)}) &= \frac{Pr[X_i^S = x_i^s \cap X_i^B = x_i - x_i^S;\alpha^{(j)}]}{Pr[X_i = x_i;\alpha^{(j)}]} \\ &= \binom{x_i}{x_i^s} \left(\frac{\alpha^{(j)}\mu_i}{\alpha^{(j)}\mu_i + \lambda_0}\right)^{x_i^S} \left(1 - \frac{\alpha^{(j)}\mu_i}{\alpha^{(j)}\mu_i + \lambda_0}\right)^{x_i - x_i^S}. \end{aligned} \tag{4.13}$$

Hence, the conditional expectation is

$$\mathbb{E}\left[X_i^S|x_i;\alpha^{(j)}\right] = \frac{x_i\alpha^{(j)}\mu_i}{\alpha^{(j)}\mu_i + \lambda_0}. \tag{4.14}$$

To calculate the second conditional expectation, $\mathbb{E}\left[Y_i^S|x_i;\alpha^{(j)}\right]$, similar steps are taken. The PMF of Y_i^S conditioned on the event $X_i = x_i$ is

$$\begin{aligned} f_{Y_i^S}(y_i^S|x_i;\alpha^{(j)}) &= \frac{Pr[Y_i^S = y_i^s \cap Y_i^B \geq T - y_i^S;\alpha^{(j)}]}{Pr[Y_i \geq T;\alpha^{(j)}]} \\ &= \frac{\left[e^{-\alpha^{(j)}\mu_i}(\alpha^{(j)}\mu_i)^{y_i^S}/y_i^S\right]\left[1 - F(T - y_i^S - 1;\lambda_0)\right]}{1 - F\left(T - 1;\alpha^{(j)}\mu_i + \lambda_0\right)}, \end{aligned} \tag{4.15}$$

where $F(A;\lambda)$ is the CDF of a Poisson distribution evaluated at A and parameterized by the mean λ.

We must evaluate the mean of this conditional distribution:

$$\mathbb{E}\left[Y_i^S|x_i;\alpha^{(j)}\right] = \sum_{y_i^S=0}^{\infty} y_i^S f_{Y_i^S}(y_i^S|x_i;\alpha^{(j)}). \tag{4.16}$$

This infinite sum does not appear to have a closed-form expression. To develop intuition on how many terms need to be included for reasonable results, we evaluated the terms of (4.16) for the situations shown in Table 4.1, which involve varying $\alpha\mu_i$, T and λ_0.

Table 4.1: Table of scenarios for conditional expectation evaluation.

Number	Description
1	$\alpha\mu_i = 0$
2	no saturation
3	Low noise, entirely saturated, low photon count
4	Low noise, entirely saturated, high photon count
5	High noise, entirely saturated, low photon count
6	High noise, entirely saturated, high photon count
7	Low noise, partly saturated, low photon count
8	Low noise, partly saturated, high photon count
9	High noise, partly saturated, low photon count
10	High noise, partly saturated, high photon count
11	Moderate noise, partly saturated, low photon count
12	Moderate noise, partly saturated, high photon count

For each of these situations, we plotted the individual terms in the summation of (4.16) for y_i^S ranging from 0 to 150. These plots include the multiplication by y_i^S needed to compute the expectation; hence they should not be interpreted as probability distribution functions. Figure 4.1 shows the plot for the low noise, partial saturation, low photon count case. The remaining plots are provided in Appendix D.

In cases where parameters are set such that a given sample is almost guaranteed to saturate, the conditional expectation for that sample becomes T or very close to T. As will be seen in Section 4.4, this has an interesting effect on the overall EM algorithm when the entire received signal has been saturated.

For the remainder of the cases, the terms of (4.16) are only greater than 1×10^{-6} for a bounded set of y_i^S. Furthermore, the conditional probability distribution (4.15) over this bounded set sums to one (within usual floating point precision). In our experiments, we only sum the first 150 terms of (4.16); this is more than enough to encapsulate the bulk of the summation.[2]

[2]The unimodality of the plots in Appendix D suggest that one could readily formulate heuristics for automatically choosing more reasonable upper limits. This might involve a trade-off between the time required to determine the upper limit, versus more brute-force evaluation. More clever approaches, such as saddle-point approximations [23], might also be feasible. We leave these as avenues for future work.

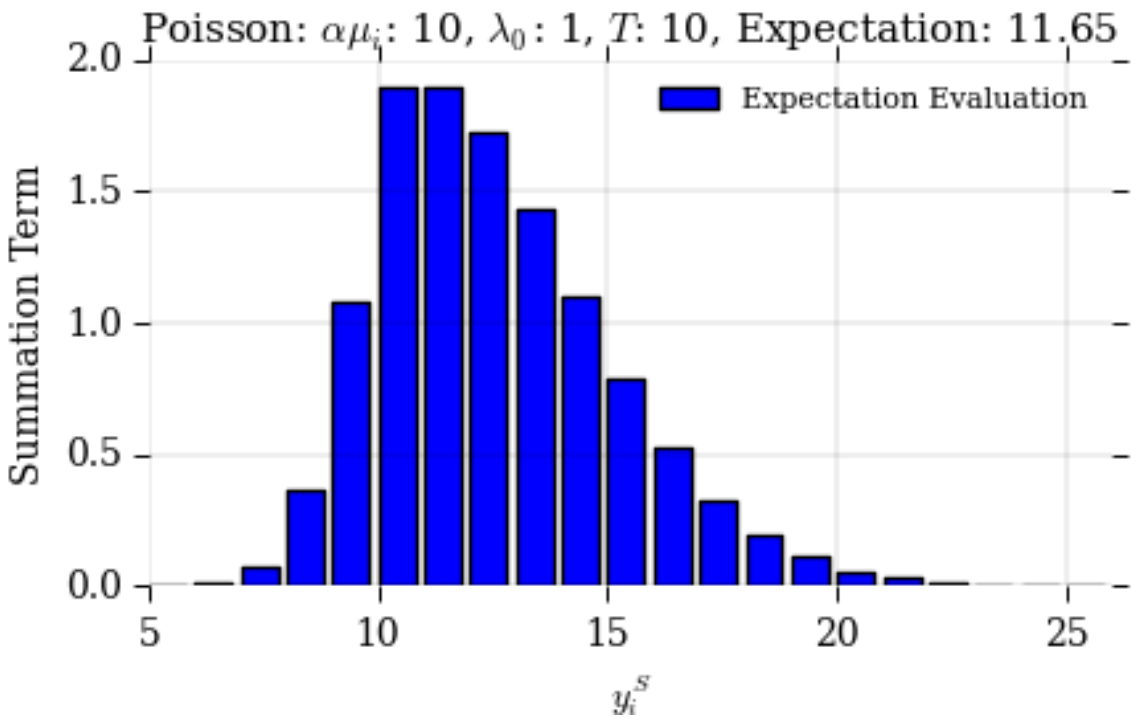

Figure 4.1: Individual terms of summation for computation of censored Poisson conditional expectation with low noise, partially saturated, and low photon count data.

4.3 EM for Gaussian Data

The derivation of the EM algorithm for the Gaussian approximation follows the derivation for the Poisson case outlined previously. This section derives an EM algorithm for estimating the scaling factor of the received OTS when it is modeled as a realization of a Gaussian random vector.

The incomplete-data loglikelihood of the observed data for this new model is

$$\begin{aligned} \ell^I(\alpha|x) &= \sum_{i=1}^{n} h_i(t), \text{ where } \lambda_i = \alpha\mu_i + \lambda_0 \text{ and} \\ h_i(t) &= \begin{cases} -\frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln(\lambda_i) - \frac{(x_i-\lambda_i)^2}{2\lambda_i} & x_i < T \\ \ln\left(1 - \frac{1}{\sqrt{2\pi\lambda_i}}\int_{-\infty}^{T} e^{-\frac{(t-\lambda_i)^2}{2\lambda_i}} dt\right) & x_i \geq T \end{cases}. \end{aligned} \tag{4.17}$$

The complete data again incorporates the saturation effects (i.e. the binning of censored and uncensored data and the random variable Y), as well as the signal decomposition

illustrated in (4.7) and (4.8). The complete loglikelihood is

$$\begin{aligned}
&\ell^C(\alpha|X^S, X^B, Y^S, Y^B) \\
&\quad = -\frac{N}{2}\ln(\alpha) - \frac{1}{2}\alpha\sum_{i=1}^{N}\mu_i \\
&\quad - \frac{1}{2\alpha}\left(\sum_{i\in\mathcal{N}}\frac{(X_i^S)^2}{\mu_i} + \sum_{i\in\mathcal{C}}\frac{(Y_i^S)^2}{\mu_i}\right),
\end{aligned} \tag{4.18}$$

where terms lacking α have been left absent in anticipation of the maximization step.

By taking the expectation of (4.18) conditioned on the received data x and $\alpha^{(j)}$, the estimate of α from iteration j, we calculate $\mathcal{Q}$ for iteration $j+1$ for the Gaussian distribution:

$$\mathcal{Q}(\alpha|\alpha^{(j)}) = -\frac{N}{2}\ln(\alpha) - \frac{1}{2}\alpha\sum_{i=1}^{N}\mu_i - \frac{C}{2\alpha}, \tag{4.19}$$

where

$$C = \sum_{i\in\mathcal{N}}\frac{\mathbb{E}\left[(X_i^S)^2|x_i;\alpha^{(j)}\right]}{\mu_i} + \sum_{i\in\mathcal{C}}\frac{\mathbb{E}\left[(Y_i^S)^2|x_i;\alpha^{(j)}\right]}{\mu_i}. \tag{4.20}$$

The x_i to the right of the vertical bars is shorthand for the event $X_i = x_i$.

Maximizing (4.19) over α for the M step of our EM algorithm is easily performed by taking the derivative of (4.19) with respect to α and setting equal to zero, resulting in the iteration

$$\alpha^{(j+1)} = \frac{-N + \sqrt{N^2 + 4C\sum_{i=1}^{N}\mu_i}}{2\sum_{i=1}^{N}\mu_i}. \tag{4.21}$$

The algorithm terminates when $\alpha^{(j+1)}$ and $\alpha^{(j)}$ become sufficiently close.

As in the Poisson case, C is complicated to evaluate. We shall begin with the first term in (4.20): $\mathbb{E}\left[(X_i^S)^2|x_i;\alpha^{(j)})\right]$. Since X^S and X are jointly Gaussian, the conditional PDF

of X_i^S conditioned on the event $X_i = x_i$ is

$$\begin{aligned} f_{X_i^S}(x_i^S|x_i;\alpha^{(j)}) &= \frac{f_{X_i^S,X_i}(x_i^S,x_i;\alpha^{(j)})}{f_{X_i}(x_i;\alpha^{(j)})} \\ &= \frac{\exp\left\{\frac{-\left[x_i^S-\frac{\alpha^{(j)}\mu_i x_i}{\alpha^{(j)}\mu_i+\lambda_0}\right]^2}{2\alpha^{(j)}\mu_i\left(1-\frac{\alpha^{(j)}\mu_i}{\alpha^{(j)}\mu_i+\lambda_0}\right)}\right\}}{\sqrt{2\pi\alpha^{(j)}\mu_i\left(1-\frac{\alpha^{(j)}\mu_i}{\alpha^{(j)}\mu_i+\lambda_0}\right)}}, \end{aligned} \tag{4.22}$$

and the conditional expectation is $\mathbb{E}\left[(X_i^S)^2|x_i;\alpha^{(j)})\right] =$

$$\alpha^{(j)}\mu_i - \frac{(\alpha^{(j)}\mu_i)^2}{\alpha^{(j)}\mu_i+\lambda_0} + \frac{(\alpha^{(j)}\mu_i x_i)^2}{(\alpha^{(j)}\mu_i+\lambda_0)^2}. \tag{4.23}$$

The evaluation of the second conditional expectation, $\mathbb{E}\left[(Y_i^S)^2|x_i;\alpha^{(j)}\right]$, is similar to that of the Poisson case. The PDF of Y_i^S conditioned on the event $X_i = x_i$ is

$$\begin{aligned} &f_{Y_i^S}(y_i^S|x_i;\alpha^{(j)}) \\ &= \frac{Pr\left[Y_i^S = y_i^S \cap Y_i^B \geq T - y_i^S;\alpha^{(j)}\right]}{Pr\left[Y_i \geq T;\alpha^{(j)}\right]} \\ &= \frac{\exp\left\{-\frac{(y_i^S-\alpha^{(j)}\mu_i)^2}{2\alpha^{(j)}\mu_i}\right\}\left[1-F(T-y_i^S;\lambda_0,\lambda_0)\right]}{\sqrt{2\pi\alpha^{(j)}\mu_i}\left[1-F(T,\alpha^{(j)}\mu_i+\lambda_0,\alpha^{(j)}\mu_i+\lambda_0)\right]}, \end{aligned} \tag{4.24}$$

where $F(A;\mu,\sigma^2)$ is the Gaussian CDF with mean μ and variance σ^2 evaluated at A.

Therefore,

$$\mathbb{E}[(Y_i^S)^2|x_i;\alpha^{(j)}] = \int_0^\infty (y_i^S)^2 f_{Y_i^S}(y_i^S|x_i,\alpha^{(j)})dy_i^S. \tag{4.25}$$

This integral does not appear to have a closed-form evaluation. However, as in the Poisson case, the integrand of (4.25) is only above 1×10^{-6} for a bounded region of y_i^S, implying that we can limit the range of the integral without significant error.

To gain intuition as to the range of integration required for reasonable results, we have evaluated (4.25) for various values of $\alpha\mu_i$, λ_0, and T for the situations outlined in Table 4.1.

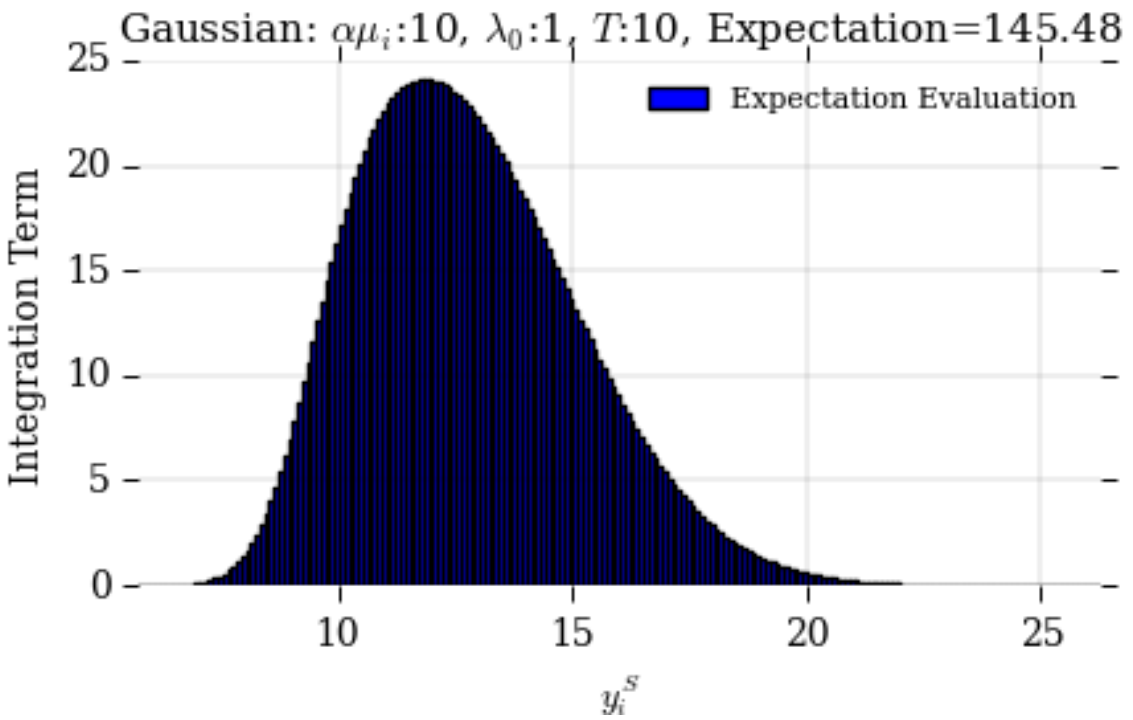

Figure 4.2: Individual terms of integration for computation of censored Gaussian conditional expectation with low noise, partially saturated, and low photon count data.

For each of these situations, we plotted the integrand of (4.25) for y_i^S ranging from 0 to 150.[3] Figure 4.2 shows the plot for the low noise, partial saturation, and low photon count situation. Appendix E shows plots for the remainder of the scenarios. To compute the integrals, we use trapezoidal integration across the range 0 to 150 with a step size of 0.1. Future work could explore more clever methods for determining an upper bound for the integration.

4.4 Results and Conclusion

This section documents the benefit of incorporating knowledge of saturation into the estimation algorithms. We begin by evaluating a single example of each of the situations outlined in Table 4.1. For each situation, we include a plot of the original intensity and the received signal (a random realization, including censoring, of a PPP), as well as recon-

[3]These plots include the y_i^S needed in the computation of an expectation, so they should not be interpreted as probability densities.

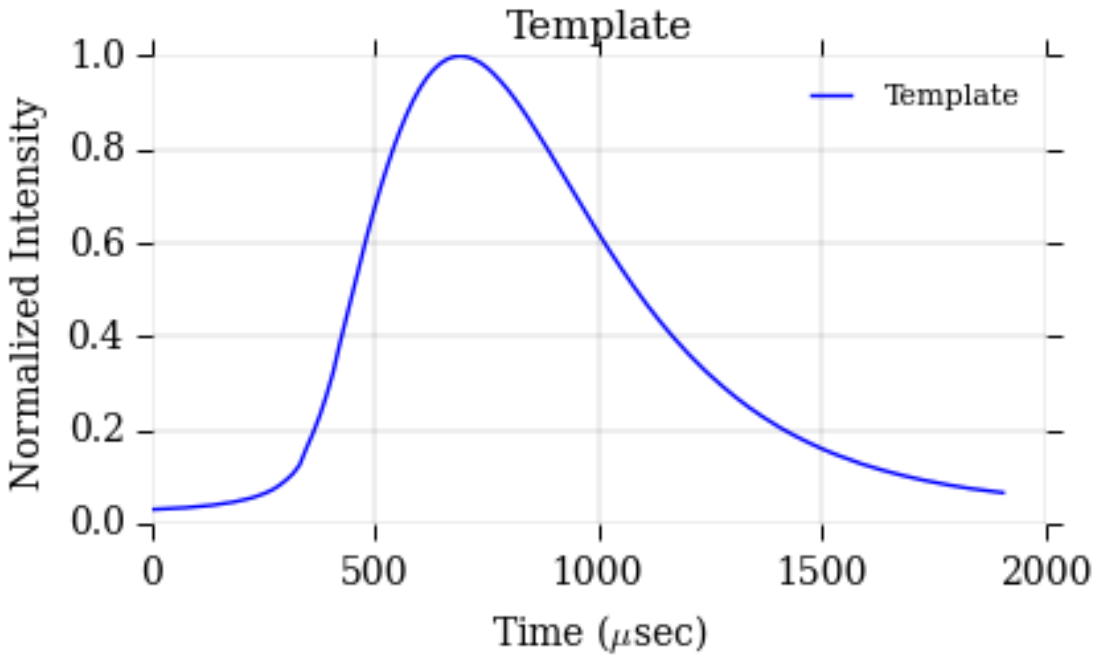

Figure 4.3: Template of a lightning strike (μ).

structed signals using α estimates from our Poisson and Gaussian EM algorithms.

Our simulations employ the OTS template displayed in Fig. 4.3. This template was generated as explained in Chapter 2 and represents a lightning strike as observed from a satellite orbiting Earth's surface.

If no saturation occurs, both algorithms estimate α within a small error. Since no time samples are saturated, the second conditional expectation in both EM algorithms, $\mathbb{E}\left[Y_i^S | x_i; \alpha^{(j)}\right]$, is absent. Figure 4.4 shows an example of an unsaturated received signal and its corresponding reconstructions. More examples are given in Appendix F.

In extreme cases where the *entire signal* has undergone saturation, our algorithms perform quite differently. This can happen when the threshold of the sensor is lower than the noise floor. Almost all information has been lost to saturation. The Poisson EM algorithm estimates α to be the threshold T. In contrast, the Gaussian EM algorithm converges to an α estimate of zero.[4] An example of this can be seen in Fig. 4.5, which corresponds to the

[4]In the case of the Gaussian EM algorithm, we terminated the iterative algorithm if $\alpha^{(j+1)}$ becomes zero. We do this to avoid a divide by zero error when calculating (4.25).

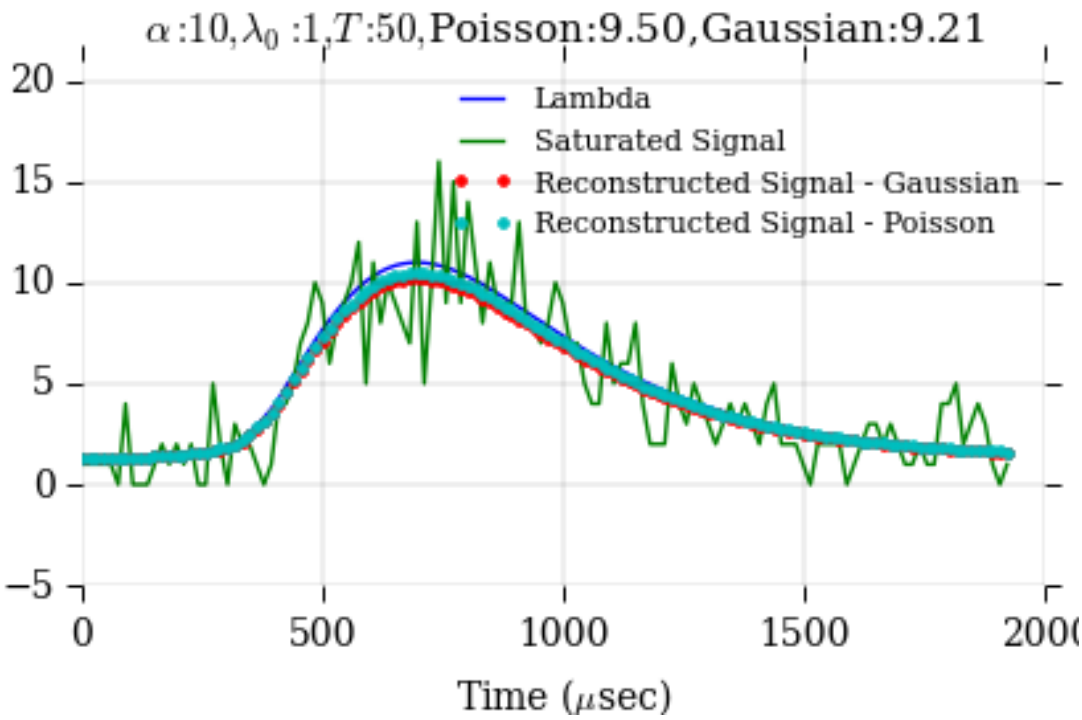

Figure 4.4: Signal reconstruction using α estimation algorithms derived from censored Poisson and Gaussian distributions on an unsaturated signal.

high noise, entirely saturated, high photon count case. We conjecture that the Gaussian EM algorithm interprets highly saturated data as mostly arising from background noise.

If we imagine starting with the fully saturated case and then gradually lowering the threshold, the received signal begins to contain more information as the threshold decreases towards the noise floor. Portions of the signature where the signature is higher tend to remain saturated, but portions where the signature is lower present a combination of saturated and unsaturated samples, giving the appearance of a flat ground with occasional dips, as illustrated in Fig. 4.6. Here, the threshold acts in such a way that the only uncensored time samples are below the noise floor λ_0, and algorithms still react in strange ways. The Poisson EM algorithm generally slightly overestimates α, but the Gaussian EM algorithm again converges to zero. As in the previous case of full saturation, it appears as though this amount of saturation fools the Gaussian EM algorithm into believing that it has received no signal.

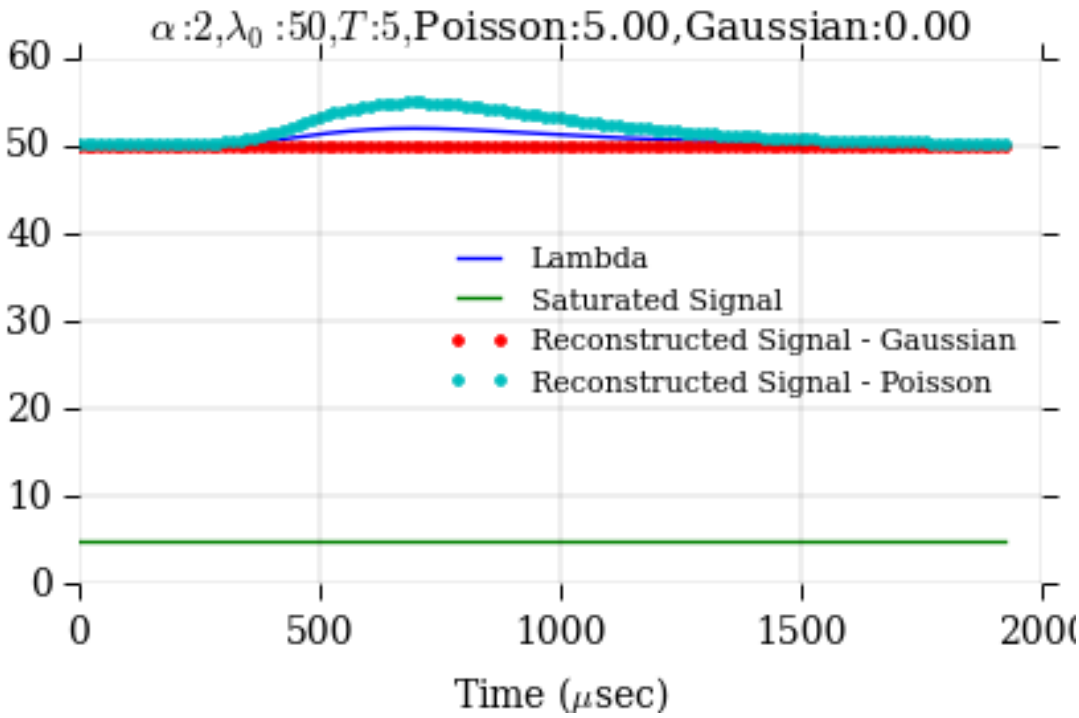

Figure 4.5: Signal reconstruction using α estimation algorithms derived from censored Poisson and Gaussian distributions on high noise, entirely saturated, high photon count data.

Finally, in the vast majority of cases where only a portion of the signal is saturated, both algorithms excel. Figure 4.7 shows an extreme case in which the signal strength is strong, the noise is low, and the threshold is low but still above the noise floor. Both algorithms accurately estimate α. Figure 4.8 displays an example in which only the higher portion of the received signal with moderate noise has undergone saturation. Both algorithms again accurately estimate α.

The remainder of the scenarios and their performance can be seen in Appendix F.

The rest of this section describes Monte Carlo simulations in which we set $T = 10$ and $\lambda_0 = 1$ and sweep α from 1 to 25, running 1000 MC iterations for each α.

To explore whether the increase in accuracy is worth the added complexity of informing the algorithm about saturation, we ran a MC simulation comparing each of our EM algorithms with EM algorithms that do not take into account saturation. The authors of [20] mention an EM algorithm for a Poisson signature with a scaling factor and additive

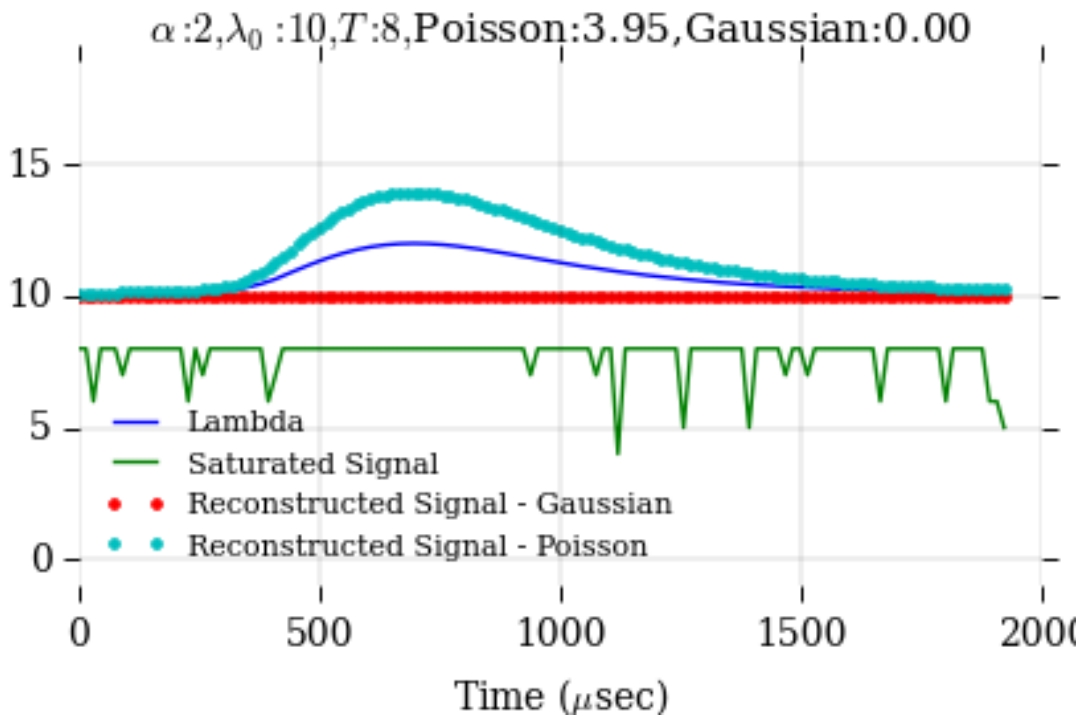

Figure 4.6: Signal reconstruction using α estimation algorithms derived from censored Poisson and Gaussian distributions on high noise, partially saturated, low photon count data.

background counts:

$$\alpha^{(j+1)} = \frac{1}{\sum_{i=1}^{n} \mu_i} \left(\sum_{i=1}^{n} \frac{x_i \mu_i \alpha^{(j)}}{\alpha^{(j)} \mu_i + \lambda_0} \right). \tag{4.26}$$

This EM algorithm is ignorant of the effect of saturation. Note that our Poisson EM algorithm (4.12) reduces to (4.26) if the received signal undergoes no saturation.

Figure 4.9 shows plots of the standard deviations of the results of two algorithms.

As expected, our censored Poisson EM algorithm performs identically to the uncensored version for small values of α, where no saturation occurs. However, as more and more samples become saturated due to the increase in α, the censored Poisson EM algorithm performs increasingly better in comparison. The figure is cropped to highlight details in the transition region. The standard Poisson EM algorithm yielded a standard deviation of 10.97 for an α value of 25, compared with a standard deviation of 1.48 resulting from the

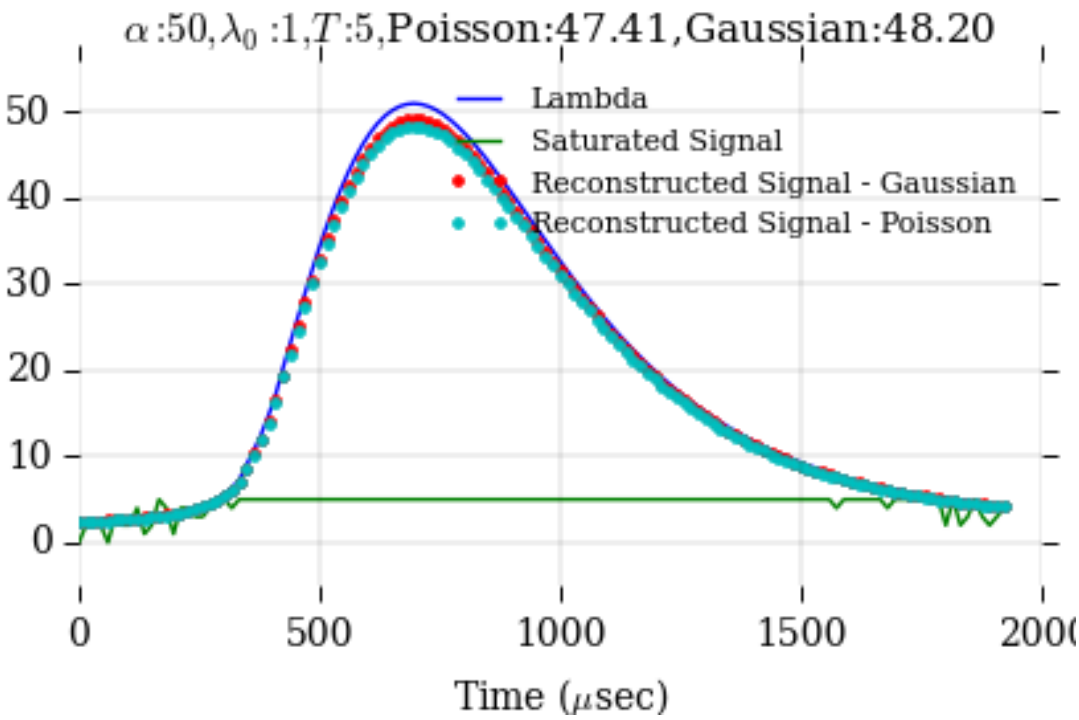

Figure 4.7: Signal reconstruction using α estimation algorithms derived from censored Poisson and Gaussian distributions on low noise, mostly saturated, high photon count data.

censored Poisson EM algorithm.

Similarly, we compare our censored Gaussian EM algorithm with a similar algorithm that does not incorporate the saturation effect. By formulating the "complete" data of the EM algorithm for the Gaussian likelihood taking into account the signal-plus-noise model of (4.7), the following Gaussian EM algorithm can be derived:

$$\alpha^{(j+1)} = -\frac{N}{2\sum_{i=1}^{n}\mu_i} + \frac{1}{2}\sqrt{\frac{N^2}{\left(\sum_{i=1}^{n}\mu_i\right)^2} + \frac{4\alpha^{(j)}}{\sum_{i=1}^{n}\mu_i}\Xi} \tag{4.27}$$

where

$$\Xi = N - \alpha^{(j)}\sum_{i=1}^{n}\frac{\mu_i}{\alpha^{(j)}\mu_i + \lambda_0} + \alpha^{(j)}\sum_{i=1}^{n}\frac{\mu_i x_i^2}{\left(\alpha^{(j)}\mu_i + \lambda_0\right)^2}. \tag{4.28}$$

The derivation follows along the lines of a simplification of Section 4.3; we omit the details for brevity. The censored Gaussian EM algorithm (4.21) reduces to (4.27) when the received signal does not undergo saturation.

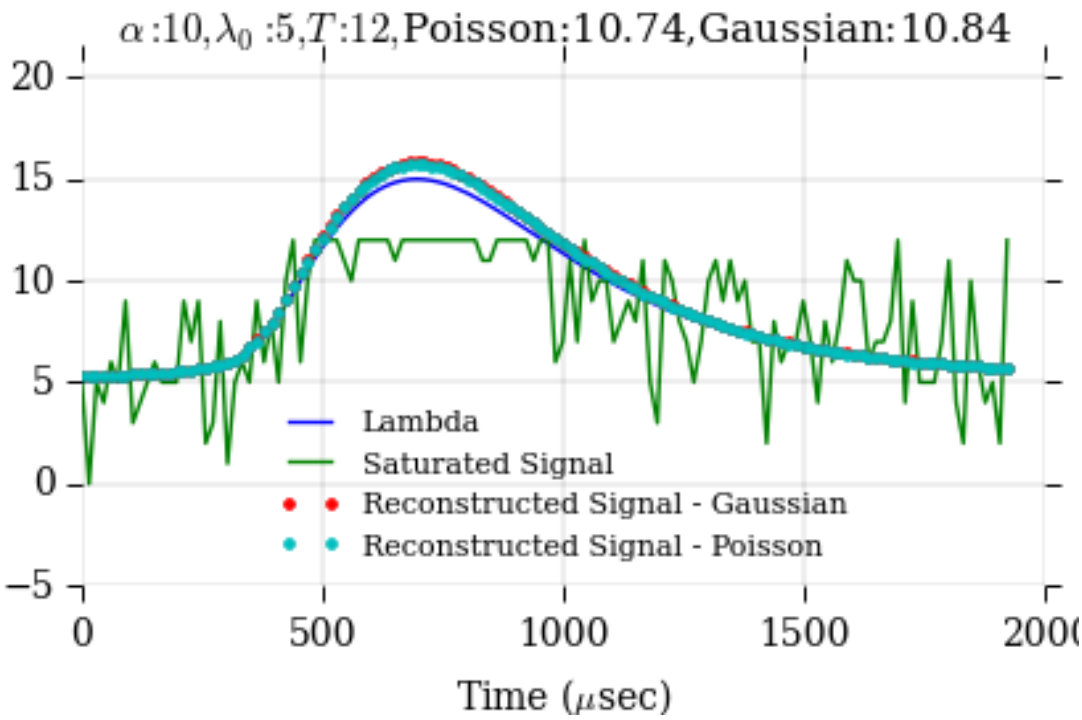

Figure 4.8: Signal reconstruction using α estimation algorithms derived from censored Poisson and Gaussian distributions on moderate noise, partially saturated, low photon count data.

Figure 4.10 shows results from a MC exploration of the Gaussian algorithms.

As in the case of the Poisson EM algorithm, both algorithms perform identically for small values of α. However, as α approaches 10 and above, the saturation effect comes into play, and the censored Gaussian EM algorithm does significantly better. At $\alpha = 25$, the uncensored EM algorithm has a standard deviation of 10.52, while the censored EM algorithm has a standard deviation of only 1.64.

Finally, we would like to compare our two censored algorithms against each other. The Monte Carlo results shown in Fig. 4.11 illustrate that the censored Poisson EM algorithm yields a slightly smaller standard deviation than the censored Gaussian EM algorithm.

4.4.1 Consequences for Time Delay Estimation

If we incorporate the α estimation algorithms from this chapter back into our TOA estimation procedure from Chapter 3, we have ML estimators for τ and α that explicitly take

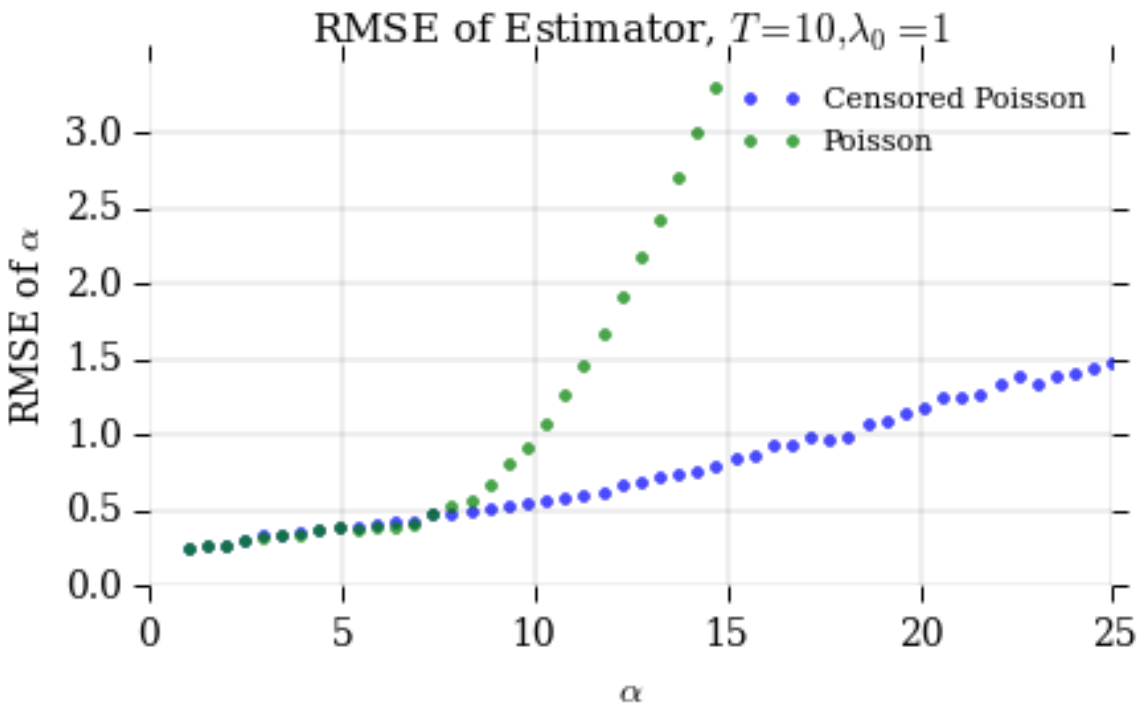

Figure 4.9: MC simulation comparing standard Poisson EM algorithm and censored Poisson EM algorithm.

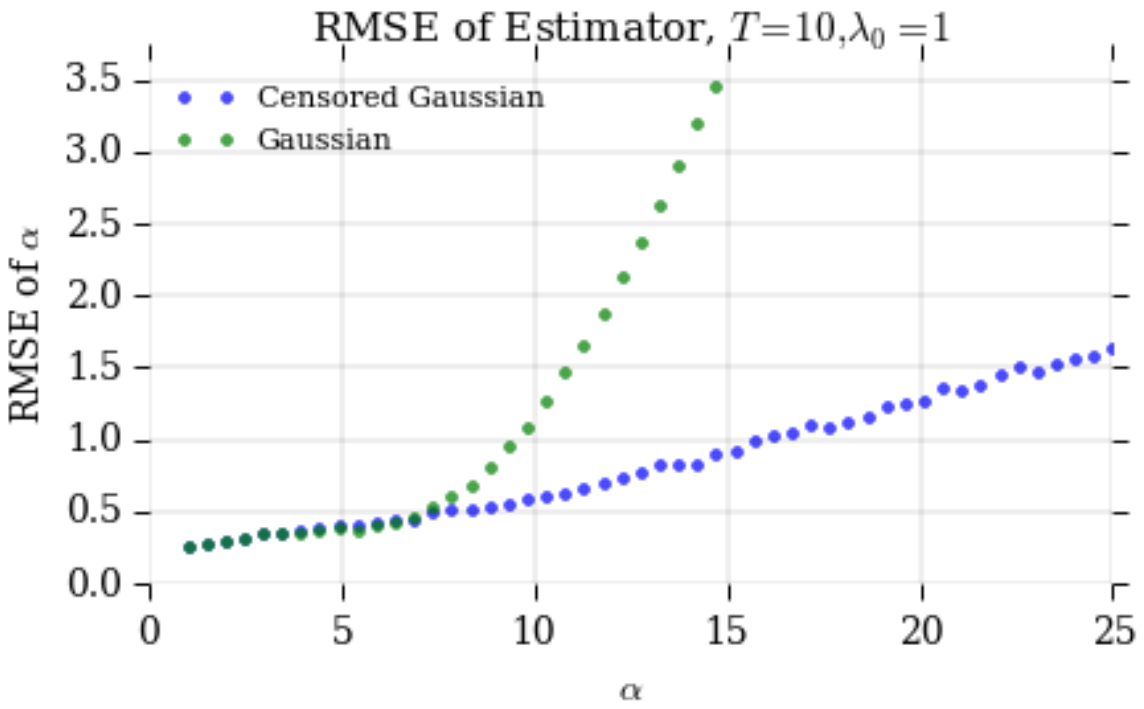

Figure 4.10: MC simulation comparing standard Gaussian EM algorithm and censored Gaussian EM algorithm.

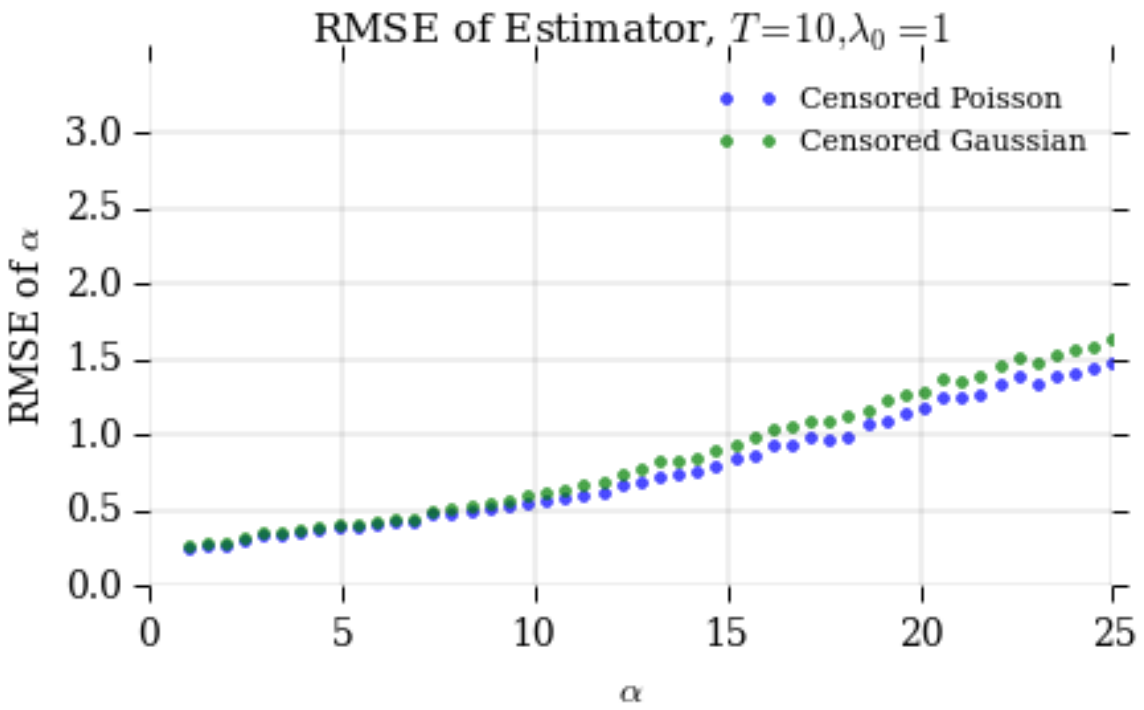

Figure 4.11: MC simulation comparing the censored Poisson EM algorithm and the censored Gaussian EM algorithm.

saturation into account. To determine the effect of not knowing α *a priori* on TOA estimation, we ran a MC simulation comparing the estimators' performances both when α was known *a priori*, as well as when it must be co-estimated. Figure 4.12 shows the results of one such simulation, which used the censored Gaussian algorithm, for $T = 10$, $\lambda_0 = 1$, and α swept from 1 to 20. It appears that the need to estimate α has very little effect on TOA estimation. While the performance of the estimators are not completely identical, they are the same for the vast majority of α values. This is likely because we can now quite accurately estimate α, compared with the situation in Chapter 3. Since the difference between estimated and true α is so minor, one would expect that the TOA estimation will also not differ by a significant margin. This behavior is consistent in all simulations run for both the censored Gaussian and censored Poisson estimators.

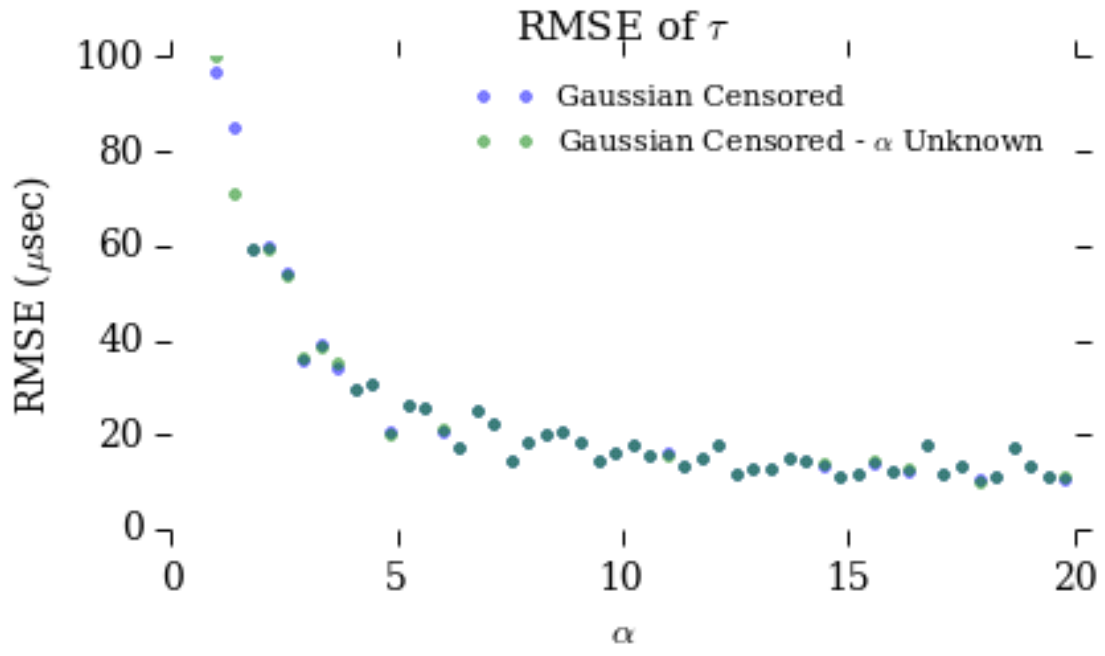

Figure 4.12: MC simulation comparing the performance of TOA estimation using the censored Gaussian estimator both knowing α a priori, as well as co-estimating α.

CHAPTER 5
CENSORED CRLB

5.1 Introduction

Chapter 3 presented theoretical lower bounds on the performance of estimators for uncensored data. We would like to derive analogous lower bounds on the performance of estimators for censored data, as seen in Chapter 4. Unfortunately, it is extremely difficult to the FIM of the censored Poisson and Gaussian loglikelihoods.

The authors of [24] explore CRLB-type lower bounds for randomly censored data. While their work is interesting, our received signals are not randomly saturated; they are saturated at a known value. The authors of [25] address the computation of a CRLB for data that has undergone Type II censoring, but that is not the saturation effect that we are exploring in this work. The authors of [26] address an iterative approach to approximating the Fisher information matrix. Although their paper did not address censoring, it lead us to the solution we implemented.

The authors of [27] derived a way to estimate the FIM of a model given an EM algorithmic framework. This chapter applies their techniques to to estimate the Fisher information for our censored Poisson and censored Gaussian α estimators, exploiting the EM framework described in Chapter 4. Inverting these estimates of the Fisher information reveals estimates of the CRLB.

This chapter employs the same notation as Chapter 4, using x for the received, saturated signal, and letting y represent the "true" received signal before saturation.

Although we attempted to adapt this method to computing the CRLBs for both τ and α, problems arose in the computation of τ. Due to the discretization of our received signal, and thus the possible values of τ, the numerical differentiation techniques we initially tried

in the computation of the CRLB for τ did not yield reasonable results. Hence, we leave performance bounds for TOA estimation for censored models as an important avenue for future work, and focus on scaling parameters.

Section 5.2 briefly outlines the method for computing FIMs in the spirit of EM algorithms as described in [27]. Section 5.3 adapts the censored Poisson EM algorithm from Chapter 4 to the FIM estimation framework. Section 5.4 does the same for the censored Gaussian algorithm. Section 5.5 presents results of these CRLBs calculations.

5.2 Summary of Methodology

This section briefly summarizes the work of [27] on estimating FIMs via "complete data" formulations. We also document the modifications to the methodology we made to adapt it to our framework.

Throughout this section, x represents the observed "incomplete" data, while y represents the complete data. The FIM of a likelihood function is

$$\mathbb{E}\left[\frac{\partial \ell(\theta|X)}{\partial \theta}\frac{\partial \ell(\theta|X)}{\partial \theta^T}\right], \tag{5.1}$$

where $\ell(\theta|X)$ is the loglikelihood. We use capital X in this instance to emphasize that the data in these expressions is treated as a random variable. Under the condition

$$\mathbb{E}\left[\frac{\partial \ell(\theta|X)}{\partial \theta}\right] = 0, \tag{5.2}$$

the expected FIM becomes

$$\mathbb{E}\left[\frac{\partial \ell(\theta|X)}{\partial \theta}\frac{\partial \ell(\theta|X)}{\partial \theta^T}\right] = -\mathbb{E}\left[\left(\frac{\partial^2 \ell(\theta|X)}{\partial \theta \partial \theta^T}\right)\right] = -\mathbb{E}\left[H(\theta|X)\right], \tag{5.3}$$

where $H(\theta|X)$ is the Hessian matrix. This condition holds true for both the Poisson and Gaussian distributions as applied to our model. Therefore, the FIM is simply the negative

expected Hessian matrix.

Let $\mathcal{Q}$ represent the expectation[1] of the complete data loglikelihood conditioned on a specific received signal, x, and the "current guess" of θ (i.e. $\alpha^{(j)}$). Here, we put "current guess" in quotes to match the exposition of the EM algorithm; in this chapter, we are not actually performing an iteration to find α, we are instead seeking the CRLB as a function of α. Taking the derivative of $\mathcal{Q}$ with respect to θ and evaluating it at $\theta^{(j)}$ yields

$$\left[\frac{\partial \mathcal{Q}\left(\theta|\theta^{(j)}\right)}{\partial \theta}\right]\Bigg|_{\theta=\theta^{(j)}} = \left[\frac{\partial \ell_o(\theta|x)}{\partial \theta}\right]\Bigg|_{\theta=\theta^{(j)}}, \tag{5.4}$$

where $\ell_o(\theta|x)$ is the "incomplete data" loglikelihood. We can compute $\mathcal{Q}$ via the E step of our EM algorithms. Furthermore, we can compute the gradient of $\mathcal{Q}$. Let

$$S\left(\theta^{(j)}|x\right) = \left[\frac{\partial \mathcal{Q}\left(\theta|\theta^{(j)}\right)}{\partial \theta}\right]\Bigg|_{\theta=\theta^{(j)}}. \tag{5.5}$$

By taking the numerical derivative of $S\left(\theta^{(j)}|x\right)$, we arrive at an *empirical* Hessian matrix for the particular data realization x.

The authors if [27] suggest using simultaneous perturbation stochastic approximation (SPSA) to compute $S\left(\theta^{(j)}|x\right)$. For this procedure, an approximation of the empirical Hessian matrix of the incomplete loglikelihood at the true value of θ for a data set x is

$$\hat{H}(\theta|x) = \frac{1}{2}\left\{\frac{\delta S}{2}\left[\Delta_1^{-1}, \Delta_2^{-1}, ..., \Delta_d^{-1}\right] + \left(\frac{\delta S}{2}\left[\Delta_1^{-1}, \Delta_2^{-1}, ..., \Delta_d^{-1}\right]\right)^T\right\}, \tag{5.6}$$

where $\delta S = S(\theta + \Delta|x) - S(\theta - \Delta|x)$ and Δ is a random perturbation vector (of size equal to the number of parameters being differentiated, which is one in our case) following the conditions outlined in [27]. Following the advice in [27], we use random perturbation vectors generated from the Bernoulli distribution.

Through MC simulations, where each simulation calculates a stochastic approximation

[1]The expectation here is with respect to the complete data Y.

of the empirical Hessian matrix $\hat{H}$ for a different realization of data, we can approximate the FIM as the negative average of these empirical Hessians.

The methodology for estimating the expected FIM from a received signal is as follows. We first choose a value for c, the value dictating the Bernoulli distribution of the perturbation vector, as well as N, the number of data sets to generate in the MC computation. For each MC run:

1. Generate a data set. For our case, this involves computing $\lambda_i = \alpha\mu_i + \lambda_0$, generating a PPP with this intensity, and saturating this signal at T.

2. Generate a perturbation vector. Since we are only deriving the CRLB for α in this chapter, our perturbation vector is of length one.

3. Apply (5.6) to calculate the estimate of the Hessian for this run.

Finally, we calculate the negative average of all $H(\theta|x)$ instances that were computed in Step 3. This is the approximation of the expected FIM. Computing the inverse of this matrix (which, in this chapter is actually a scaler) yields an approximate CRLB for α.

The following sections adapt this methodology for the censored Poisson and Gaussian estimators.

5.3 CRLB for Poisson data

This section adapts the EM algorithm presented in Chapter 4 for censored Poisson data for approximating the CRLB given the methodology summarized in Section 5.2.

The model for the received signal is identical to that of Section 4.2, based on the summation of a signal component (a realization of a PPP with intensity $\alpha\mu_i$) plus background noise (a realization of a PPP with intensity λ_0). For the purposes of this chapter, like in Chapter 4, we ignore the estimation of the TOA and assume all received signals are time-

aligned with our template, μ. Therefore, our complete-data loglikelihood is

$$\begin{aligned}\ell^C(\theta|x) = -\alpha\sum_{i=1}^{N}\mu_i - \sum_{i=1}^{N}\lambda_0 + \sum_{i\in\mathcal{N}} X_i^S \ln\left(\alpha\mu_i\right) + \sum_{i\in\mathcal{N}} X_i^B \ln\left(\lambda_0\right) \\ + \sum_{i\in\mathcal{N}} Y_i^S \ln\left(\alpha\mu_i\right) + \sum_{i\in\mathcal{N}} Y_i^B \ln\left(\lambda_0\right),\end{aligned} \tag{5.7}$$

where $\theta = [\alpha]$.

By taking the expectation of (5.7) conditioned on the received signal x and the "current estimates" for the parameters, $\theta^{(j)}$, we compute

$$\mathcal{Q}(\theta|\theta^{(j)}) = -\alpha\sum_{i=1}^{N}\mu_i + \ln(\alpha)\left(\sum_{i\in\mathcal{N}} A_i + \sum_{i\in\mathcal{C}} B_i\right), \tag{5.8}$$

where

$$A_i = \mathbb{E}\left[X_i^S|x_i;\theta^{(j)}\right] = \frac{x_i\alpha^{(j)}\mu_i}{\alpha^{(j)}\mu_i + \lambda_0} \tag{5.9}$$

and

$$B_i = \mathbb{E}\left[Y_i^S|x_i;\theta^{(j)}\right] = \sum_{y_i^S=0}^{\infty} y_i^S \frac{\left[e^{-\alpha^{(j)}\mu_i}\left(\alpha^{(j)}\mu_i\right)^{y_i^S}/y_i^S!\right]\left[1-F(T-y_i^S-1;\lambda_0)\right]}{1-F\left(T-1;\alpha^{(j)}\mu_i+\lambda_0\right)}, \tag{5.10}$$

as derived in Section 4.2.

We must take the derivate of $\mathcal{Q}$ with respect to α. As such,

$$\frac{\partial\mathcal{Q}(\theta|\theta^{(j)})}{\partial\alpha} = -\sum_{i=1}^{N}\mu_i + \frac{1}{\alpha}\left(\sum_{i\in\mathcal{N}} A_i + \sum_{i\in\mathcal{C}} B_i\right), \tag{5.11}$$

resulting in

$$S(\theta^{(j)}|x) = -\sum_{i=1}^{N}\mu_i + \frac{1}{\alpha^{(j)}}\left(\sum_{i\in\mathcal{N}} A_i + \sum_{i\in\mathcal{C}} B_i\right). \tag{5.12}$$

We plug in (5.12) to (5.6) to compute the empirical Hessian for each MC dataset.

5.4 CRLB for Gaussian data

This section adapts the EM algorithm presented in Chapter 4 for censored Gaussian data toward the approximation of the CRLB, using the methodology summarized in Section 5.2.

The complete-data loglikelihood for censored Gaussian data applied to our model is

$$\begin{aligned}\ell^C(\theta|x) = &-\frac{1}{2}\sum_{i=1}^{N}\ln(\alpha\mu_i) - \frac{1}{2}\sum_{i=1}^{N}\lambda_0 - \frac{1}{2}\sum_{i\in\mathcal{N}}^{N}\frac{\left(X_i^S\right)^2}{\alpha\mu_i} - \frac{1}{2}\sum_{i\in\mathcal{N}}^{N}\frac{\left(X_i^B\right)^2}{\lambda_0} \\ &- \frac{1}{2}\sum_{i\in\mathcal{C}}^{N}\frac{\left(Y_i^S\right)^2}{\alpha\mu_i} - \frac{1}{2}\sum_{i\in\mathcal{C}}^{N}\frac{\left(Y_i^B\right)^2}{\lambda_0} - \frac{1}{2}\sum_{i=1}^{N}\alpha\mu_i - \frac{N}{2}\lambda_0,\end{aligned} \tag{5.13}$$

where $\theta = [\alpha]$.

Taking the expectation of (5.13) conditioned on the received signal x and the "current guess" of α, namely $\alpha^{(j)}$ yields

$$\mathcal{Q}(\theta|\theta^{(j)}) = -\frac{N}{2}\ln(\alpha) - \frac{1}{2\alpha}\left(\sum_{i\in\mathcal{N}}\frac{A_i}{\mu_i} + \sum_{i\in\mathcal{C}}\frac{B_i}{\mu_i}\right) - \frac{\alpha}{2}\sum_{i=1}^{N}\mu_i, \tag{5.14}$$

where $A_i = \mathbb{E}\left[\left(X_i^S\right)^2|x_i;\theta^{(j)}\right] =$

$$\alpha^{(j)}\mu_i - \frac{\left(\alpha^{(j)}\mu_i\right)^2}{\alpha^{(j)}\mu_i + \lambda_0} + \frac{\left(\alpha^{(j)}\mu_i x_i\right)^2}{\left(\alpha^{(j)}\mu_i + \lambda_0\right)^2}, \tag{5.15}$$

and $B_i = \mathbb{E}\left[\left(Y_i^S\right)^2|x_i;\theta^{(j)}\right] =$

$$\int_0^\infty \frac{(y_i^S)^2}{\mu_i}\frac{\left[\exp\left\{-\frac{\left(y_i^S-\alpha^{(j)}\mu_i\right)^2}{2\alpha^{(j)}\mu_i}\right\}\Big/\sqrt{2\pi\alpha^{(j)}\mu_i}\right]\left[1-F(T-y_i^S;\lambda_0,\lambda_0)\right]}{1-F(T,\alpha^{(j)}\mu_i+\lambda_0,\alpha^{(j)}\mu_i+\lambda_0)}dy_i^S, \tag{5.16}$$

as derived in Section 4.3.

Taking the derivative of $\mathcal{Q}(\theta|\theta^{(j)})$ with respect to α and evaluating at $\alpha = \alpha^{(j)}$ gives

$$S(\theta^{(j)}|x) = -\frac{N}{2\alpha^{(j)}} + \frac{1}{2\left(\alpha^{(j)}\right)^2}\left(\sum_{i\in\mathcal{N}}\frac{A_i}{\mu_i} + \sum_{i\in\mathcal{C}}\frac{B_i}{\mu_i}\right) - \frac{1}{2}\sum_{i=1}^{N}\mu_i. \tag{5.17}$$

Plugging (5.17) into (5.6) yields the empirical Hessian for each MC dataset.

5.5 Results

This section presents our CRLB approximations for censored Poisson and Gaussian estimators of the scaling factor, α. For conciseness, we omit the term "approximation" in the rest of this Chapter. We first compare the censored Poisson CRLB with the performance of the censored Poisson α estimation algorithm found via a MC simulation. We then compare the censored Poisson and censored Gaussian CRLBs. The MC simulations in this section sweep α from 1 to 20 while keeping $T = 7$ and $\lambda_0 = 3$. We finally compare the uncensored and censored CRLBs.

Figure 5.1 shows the CRLB for α estimation using the censored Poisson model, as well as the performance of the same maximum-likelihood estimator found via a MC simulation. We plot the square-root of the CRLB and compare it to RMSE values from the MC simulation. As illustrated by this plot, the CRLB strictly bounds estimator performance, so the ML estimate is not efficient. The difference between the RMSE and the CRLB increases with α.

Figure 5.2 compares the computed CRLBs for the censored Poisson and censored Gaussian models. Simulations throughout the thesis tend to show that these models yield different performance; this arises because all of those MC simulations employ Poisson data. This is mirrored in the CRLBs. Future work could include simulations with data derived from a true Gaussian model.

Figure 5.3 compares the CRLBs for the censored and uncensored Poisson models. We show the uncensored CRLB for solely α estimation, as well as for co-estimation with τ.

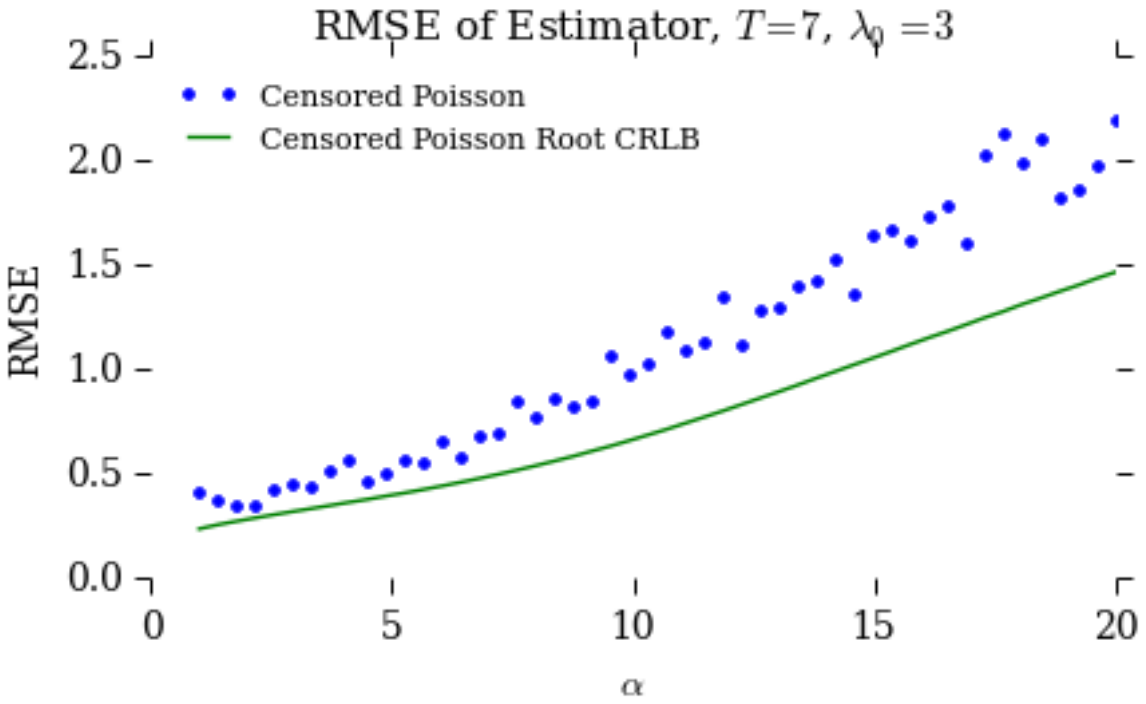

Figure 5.1: Approximation of censored Poisson CRLB for $T = 7$, $\lambda_0 = 3$.

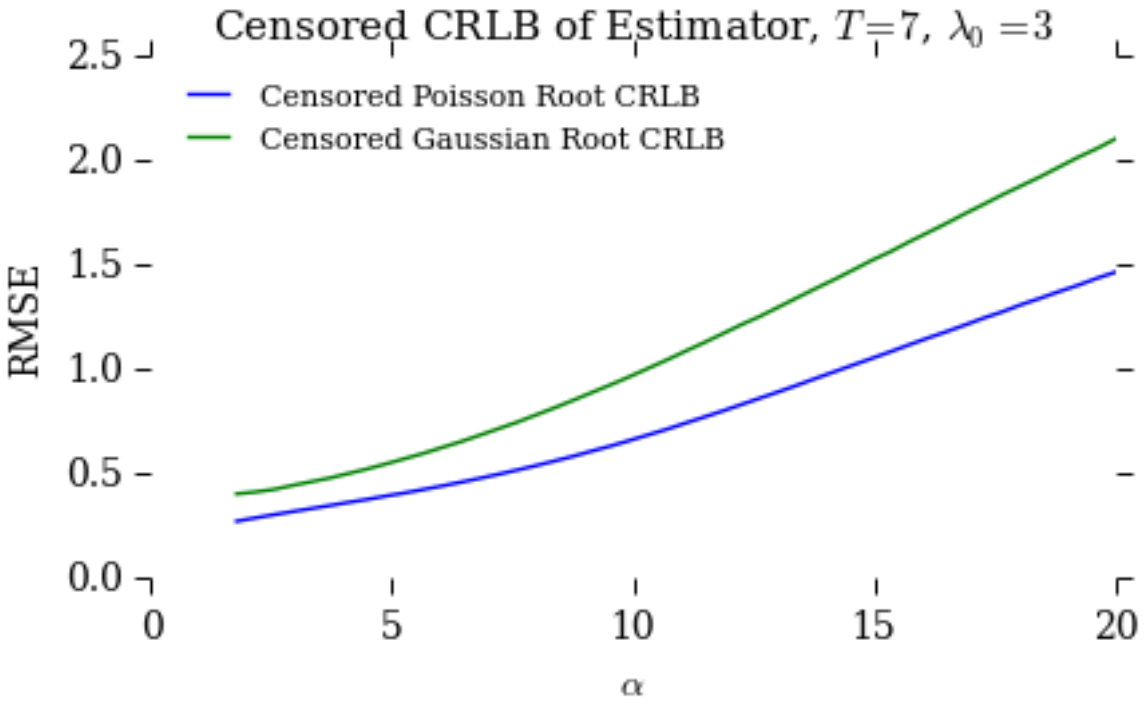

Figure 5.2: Comparison of the censored Poisson and Gaussian CRLBs. The Poisson CRLB predicts better performance than the Gaussian CRLB.

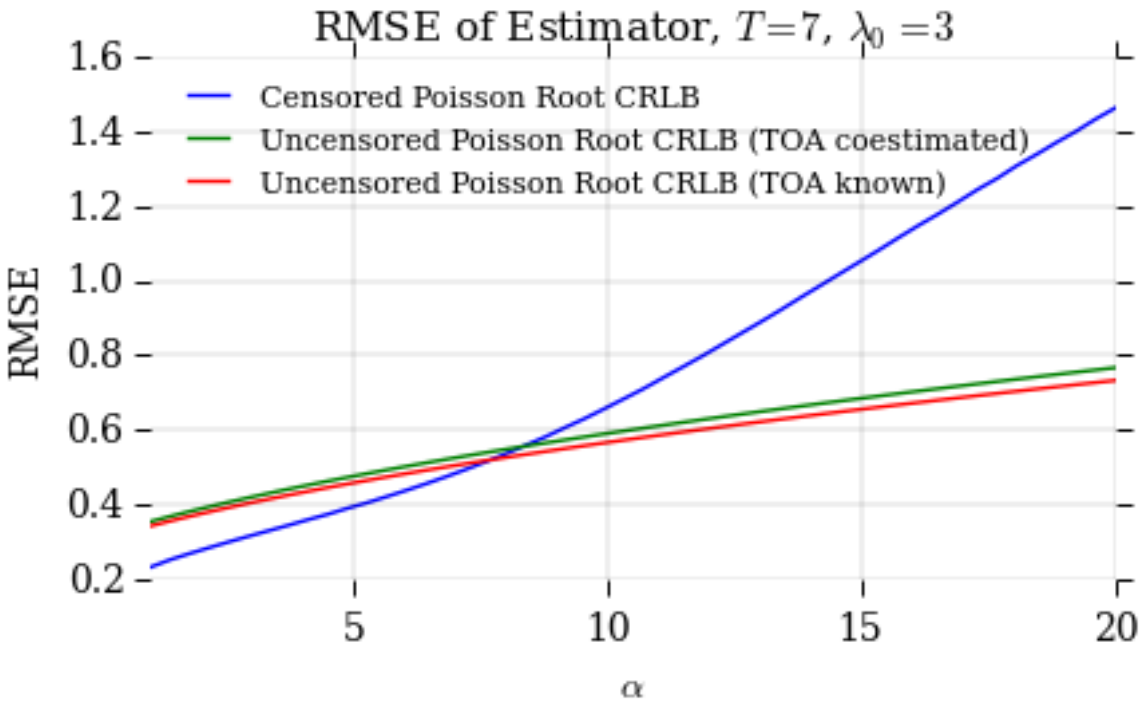

Figure 5.3: Comparison of the censored and uncensored Poisson CRLBs.

Counterintuitively, these bounds cross, suggesting that more information is available from the censored data for low α. It does not appear as though TOA co-estimation significantly effects the performance of α estimation. We predict that incorporating TOA co-estimation into the censored CRLB would not produce a significant change in the bound.

CHAPTER 6
RESULTS AND CONCLUSIONS

6.1 Introduction

This chapter contains the results of MC simulations comparing our four estimators for the TOA of a received optical signal undergoing saturation, based on uncensored and censored Poisson models, as well as censored and uncensored Gaussian models. Since the Gaussian results follow the same trends as the Poisson results, we focus on Poisson models below. The main difference between the Gaussian and Poisson algorithms is that the Poisson algorithms slightly outperform the Gaussian algorithms; recall that all test data is generated with the Poisson model. We would expect this performance gap to decrease as the number of counts increases.

This chapter concludes with suggestions for future work.

6.2 Results

This section examines two extreme-case situations for TOA estimation given an OTS that has undergone saturation. The first simulation sets $T = 4$ and $\lambda_0 = 1$, sweeping α from 1 to 20 in 50 steps. This corresponds to a sensor where at least a few time samples undergo saturation in every received signal. Furthermore, this exemplifies the "photon-starved" condition outlined in Chapter 1 that motivates the PPP model. The second simulation sets $T = 7$ and $\lambda_0 = 3$, sweeping α from 1 to 20 in 50 steps. This explores the effect of moderate saturation when some noise is present.

Figures 6.1 and 6.2 display the performance of the uncensored and censored Poisson estimators in the low threshold scenario. There is little noticeable difference in the performance of these two TOA estimators. While the censored Poisson estimator benefits from

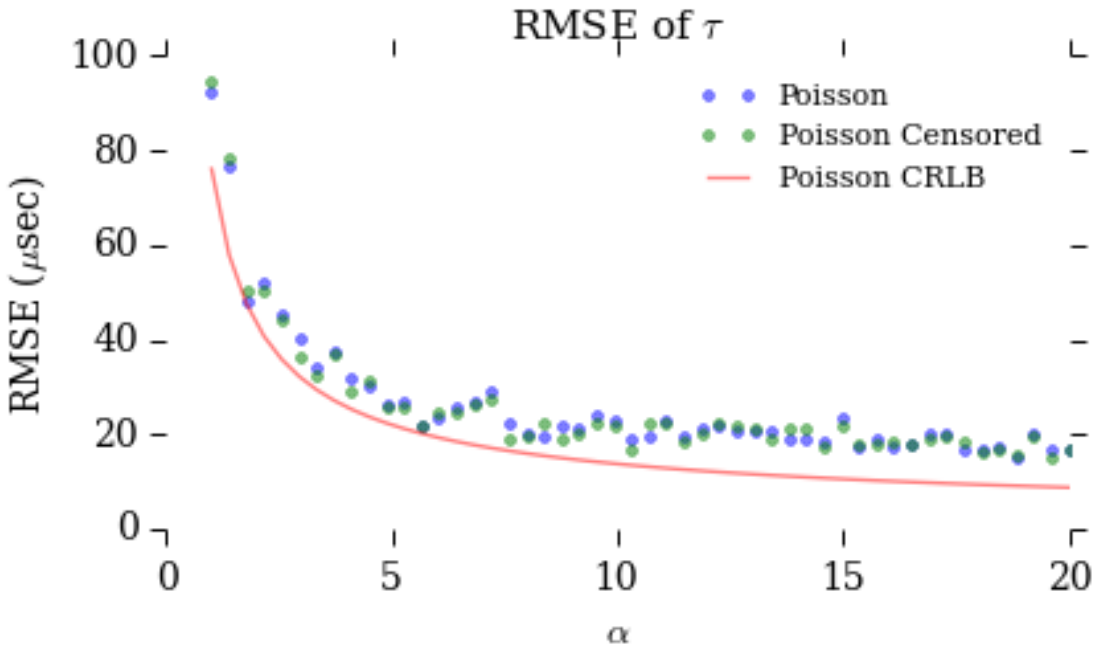

Figure 6.1: RMSE of the uncensored Poisson and censored Poisson TOA estimators from a MC simulation using $T = 4$ and $\lambda_0 = 1$.

lower RMSE in the majority of cases, the performance difference is negligible for all values of α tested. In contrast, Fig.6.2 shows that the performance of the censored Poisson α estimator significantly outperforms the uncensored estimator for all α values greater than T.

Figures 6.3 and 6.4 display the results of the simulation with $T = 7$ and $\lambda_0 = 3$, which represents a scenario with a moderate threshold but with notable background noise. Figure 6.3 shows the RMSE of the uncensored and censored Poisson estimators for τ. Here, the improvement the censored Poisson estimator provides over that of the uncensored Poisson estimator is more significant than seen in the $T = 4$, $\lambda_0 = 1$ case, although the improvement is still small.[1]

Figure 6.4 emphasizes the trend that the censored α estimator yields considerable improvement in estimating the scaling factor of the signal. The uncensored Poisson estimator

[1]We conjecture that the few points for which the uncensored algorithm appears to perform better than the censored algorithm are statistical anomalies arising from the limited number of MC runs. These anomalies are more prevalent at low α levels.

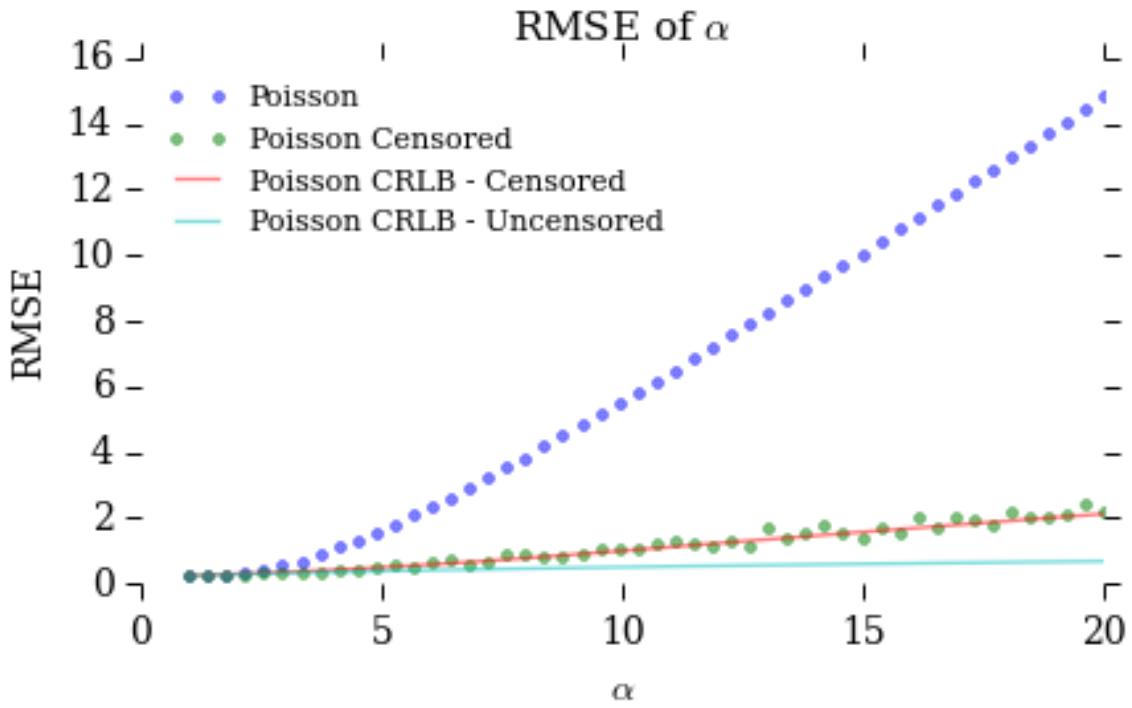

Figure 6.2: RMSE of the uncensored Poisson and censored Poisson α estimators from a MC simulation using $T = 4$ and $\lambda_0 = 1$.

again continually estimates an α value of slightly larger that T, which results in the linear increase in RMSE as seen in the figure. In contrast, the censored estimator appears to well match the CRLB for the censored Poisson distribution.

These results are consistent with our counterintuitive findings in Chapter 3: although the uncensored estimation algorithms remain ignorant of the saturation of the received signal, the α estimation algorithms can compensate by underestimating α to "fit" the saturation. Figure 6.5 illustrates this adaptation. To make this figure, we created template intensities as defined in (1.2) for α values from 1 to 20 with $\lambda_0 = 1$. We saturated these templates at different thresholds ($T = 5, 7, 10$, and 15) and applied our uncensored Poisson α estimator to them. For the purpose of making this plot, we assume τ is accurately known and does not need to be estimated. Also, the "received signal" is not a realization of a PPP; it is the actual template, saturated at a known threshold.

Figure 6.5 displays the estimated α versus the true α for several different saturation

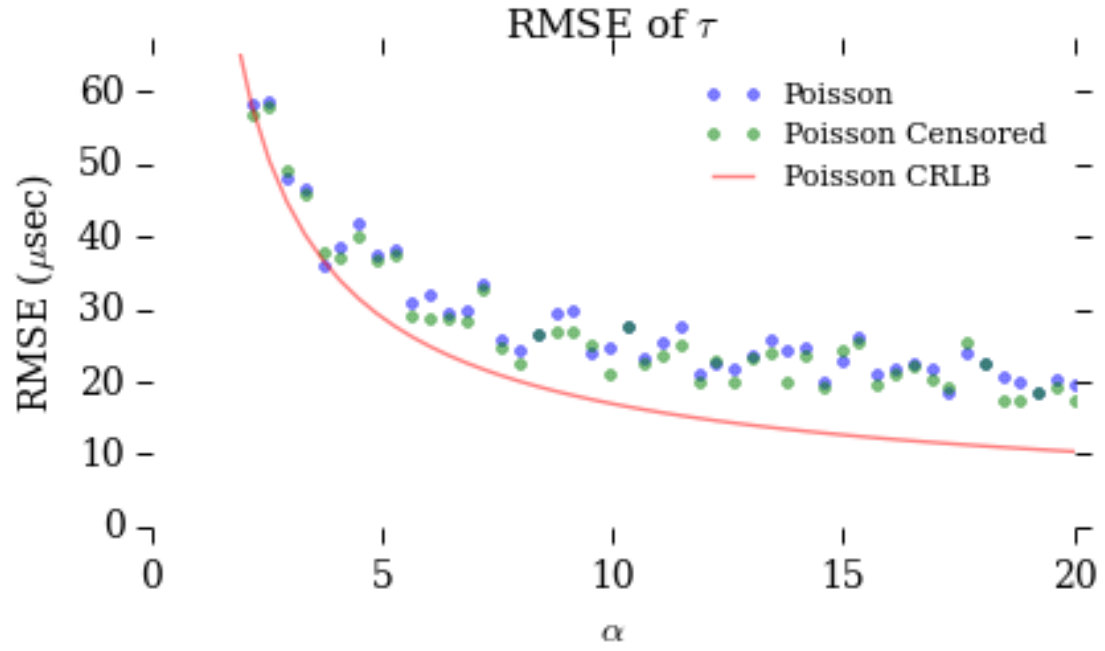

Figure 6.3: RMSE of the uncensored Poisson and censored Poisson TOA estimators from a MC simulation using $T = 7$ and $\lambda_0 = 3$.

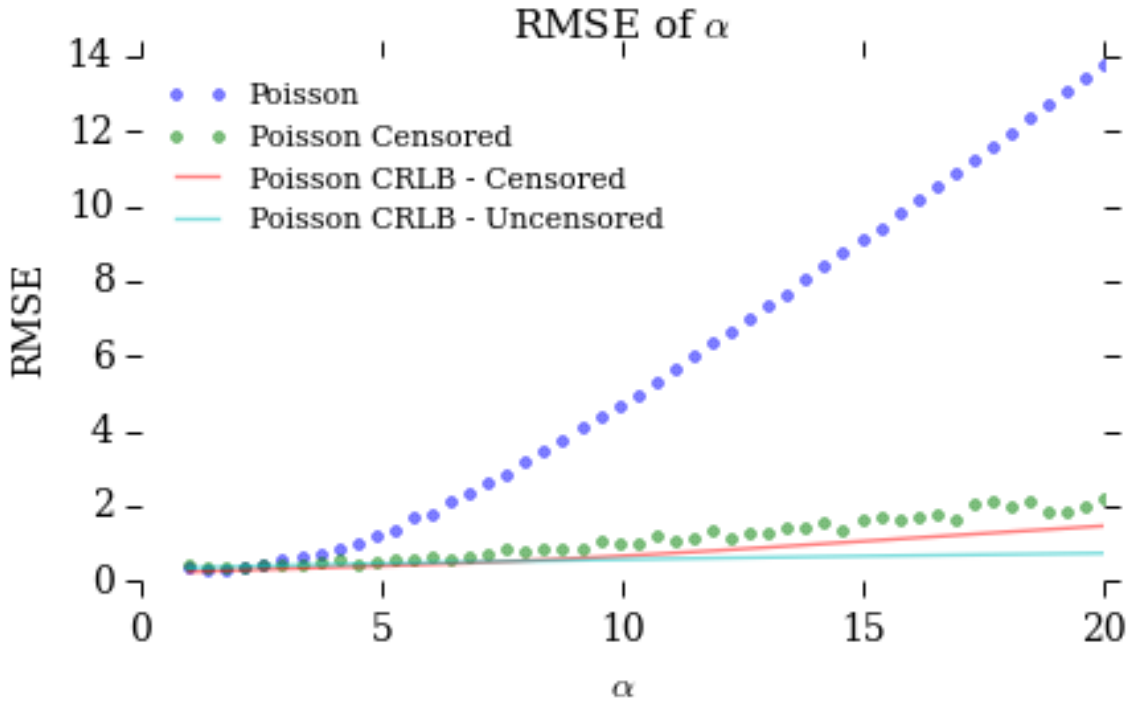

Figure 6.4: RMSE of the uncensored Poisson and censored Poisson α estimators from a MC simulation using $T = 7$ and $\lambda_0 = 3$.

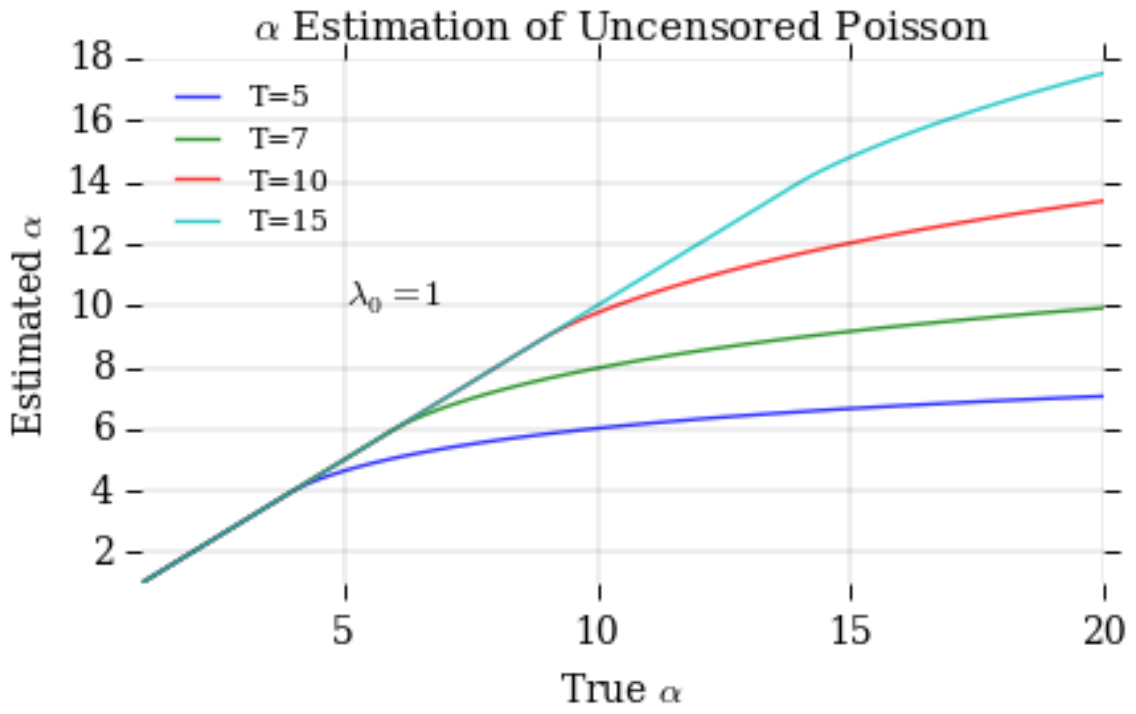

Figure 6.5: α prediction using the uncensored Poisson distribution. The inaccuracy of the α prediction is what enables the uncensored models to adapt and incorporate saturation into the estimation of the TOA, enhancing the performance of that estimation.

thresholds. For each threshold, the same pattern occurs: the uncensored EM algorithm accurately reports the true α until the true α approaches T. At this point, where the true α is just below T, the estimated curve begins to taper off, straying from the linear increase with α to something resembling a logarithmic increase with α. This underestimation of α seems to allow the uncensored models to implicitly incorporate the effects of saturation.

Figure 6.6 shows a random realization of our PPP saturated at $T = 7$, with a background level of $\lambda_0 = 3$, compared with reconstructed signals given α estimates from our censored and uncensored Poisson algorithms. The uncensored Poisson algorithm estimates α to be just slightly larger than T. This underestimation allows the "wrong" uncensored model to reasonably match the data, and appears to facilitate this "wrong" model in estimating the TOA.

Somewhat ironically, the inaccuracy of uncensored α estimation resulted in accurate TOA estimation. However, as illustrated by comparing 6.3 with 6.1, the unintentional data-

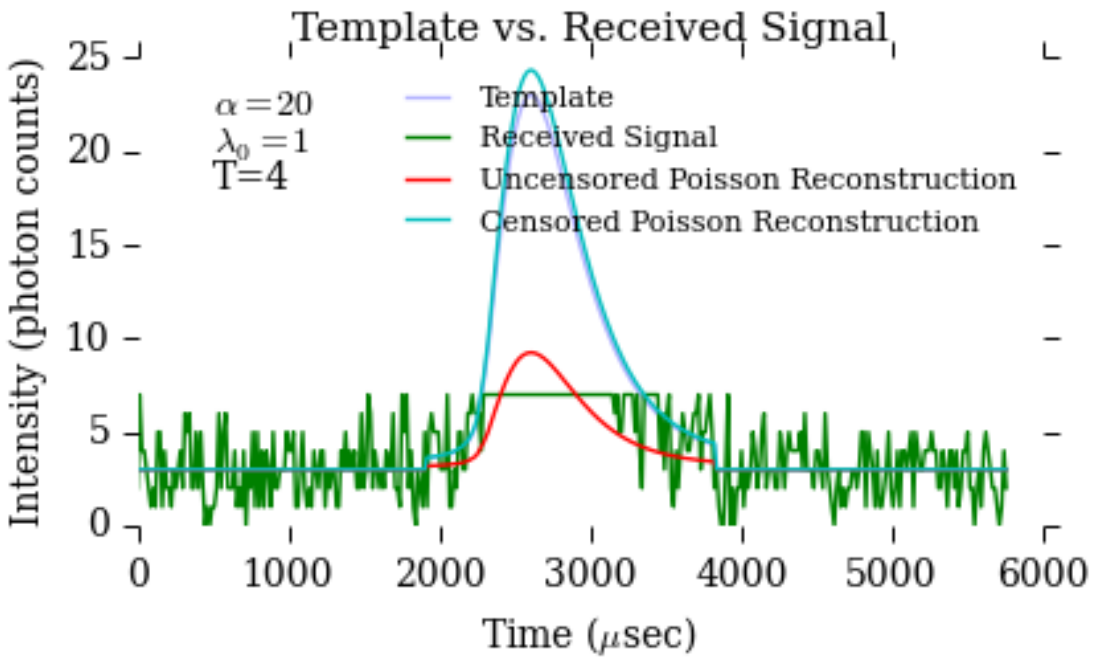

Figure 6.6: Signal reconstruction comparing the uncensored and censored Poisson α estimations. While the uncensored Poisson algorithm significantly underestimates the value of α, this allows the "ignorant" estimator to incorporate saturation into the model.

matching ability of the uncensored algorithm appears to lose its effectiveness with higher background levels. At these higher levels, we conjecture that the uncensored α estimation procedure has difficulty with simultaneously matching the time regions where the underlying template has high intensities and the time regions where the underling template has low intensities.

Our overall results suggest that if the TOA of a received saturated OTS is the only parameter of interest, the "ignorant" algorithms might be preferable because the evaluation of the conditional expectation in the censored EM algorithms are computationally expensive. However, we would be hesitate to extrapolate this suggestion to scenarios that differ from those used in our experiments. In particular, the template used throughout this thesis is relatively simple, consisting of a single hump; a more complicated structure might induce different trends, since the uncensored algorithm might have difficulty simultaneously accommodating varying intensity levels in a more complex template. Also, there may be

situations where α is an interesting parameter in its own right. For instance, meteorologists may want to characterize the intensity of lightning strikes in addition to their distances, in which case the censored model would be unquestionably worth the extra computation.

6.3 Future Work

The most obvious suggestion for future work would be to experiment with more complicated template shapes. Instead of locating lightning strikes, we could imagine locating an optical communications signal or a laser radar transmission with a complicated time-varying intensity. We suspect that more complex signals, such as sums of rectangular pulses with different magnitudes, could yield more notable performance differences between the censored and uncensored estimators.

We also suggest exploring more highly parameterized models of the received signal. This thesis explored only two parameters: the TOA, τ, and the scaling factor, α. While the TOA estimation did not seem to greatly benefit from the informed models (although the benefit increased with higher background levels), the scaling factor estimation unquestionably did. The single amplitude parameter α represented the simplest possible template variation. As shown in Chapter 2, lightning signatures can vary drastically. Some lightning profiles have a two-humped structure, in which the second hump arises from a return stroke. One could postulate a three-parameter model to characterize the amplitudes of the hump and the distance between them. For both single-strike and double-strike models, one could add additional parameters to characterize the time widths of the strikes, and/or the asymmetry of their profiles. The EM algorithms of Chapters 3 and 4 could be extended to accommodate these additional parameters, along with minimum-description-length inspired complexity penalties to avoid overfitting. These more flexible template models may benefit more strongly from properly formulated censored likelihood models.

Chapter 5 derived a CRLB for α given saturation effects but assuming a known TOA. Future work could extend this to create a joint bound including TOA estimation.

Appendices

APPENDIX A

LIGHTNING GROUPS

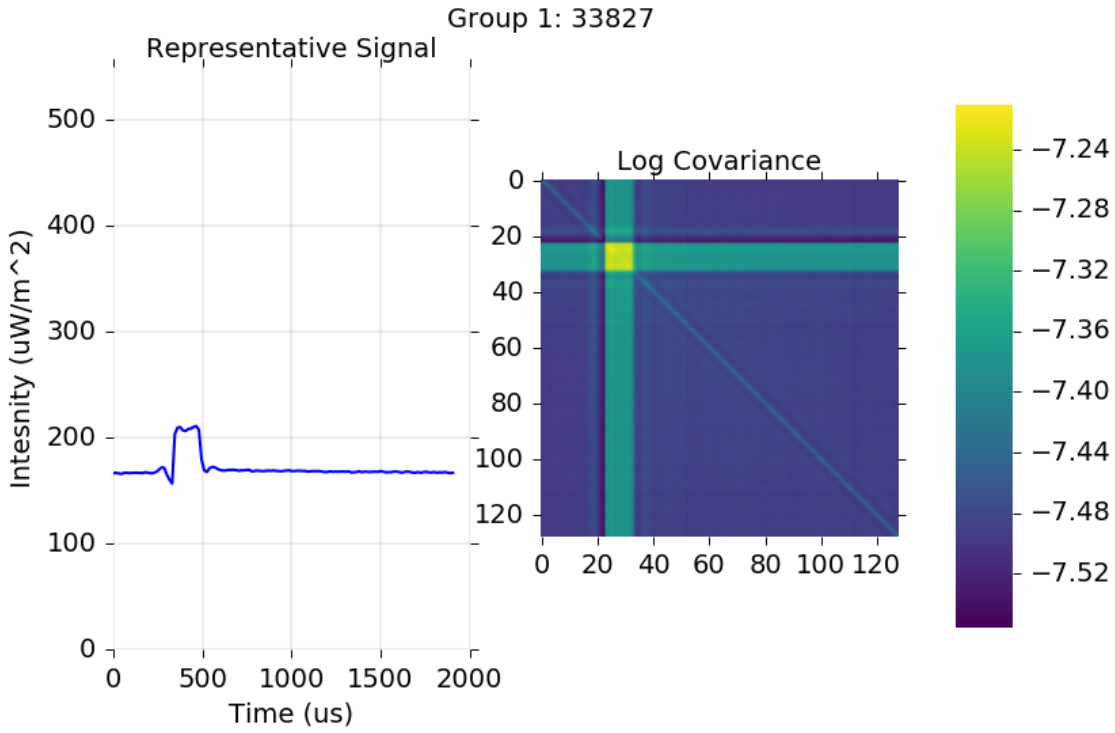

Figure A.1: Group 1 representative signal and covariance matrix.

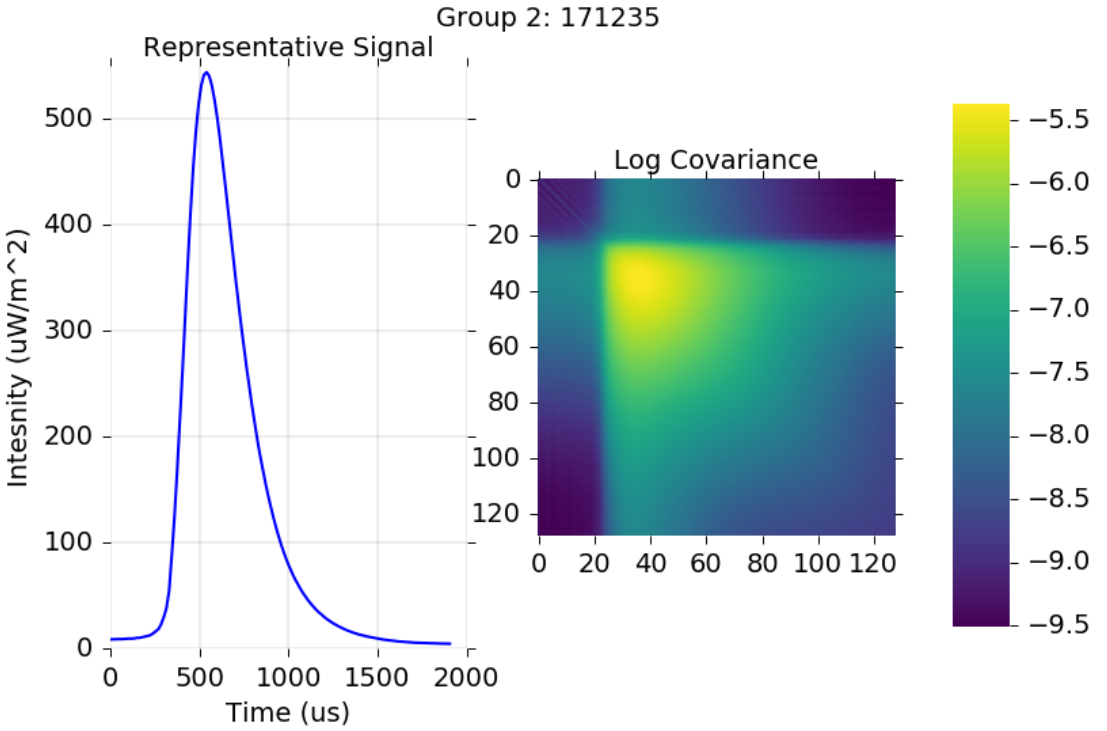

Figure A.2: Group 2 representative signal and covariance matrix.

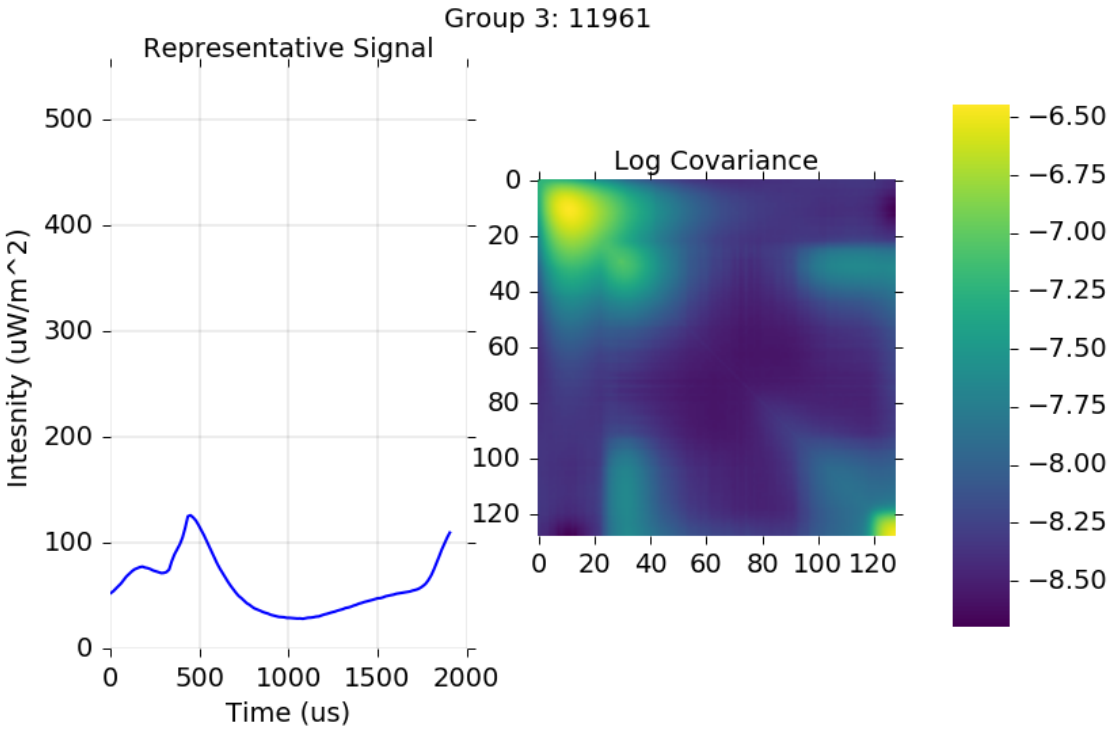

Figure A.3: Group 3 representative signal and covariance matrix.

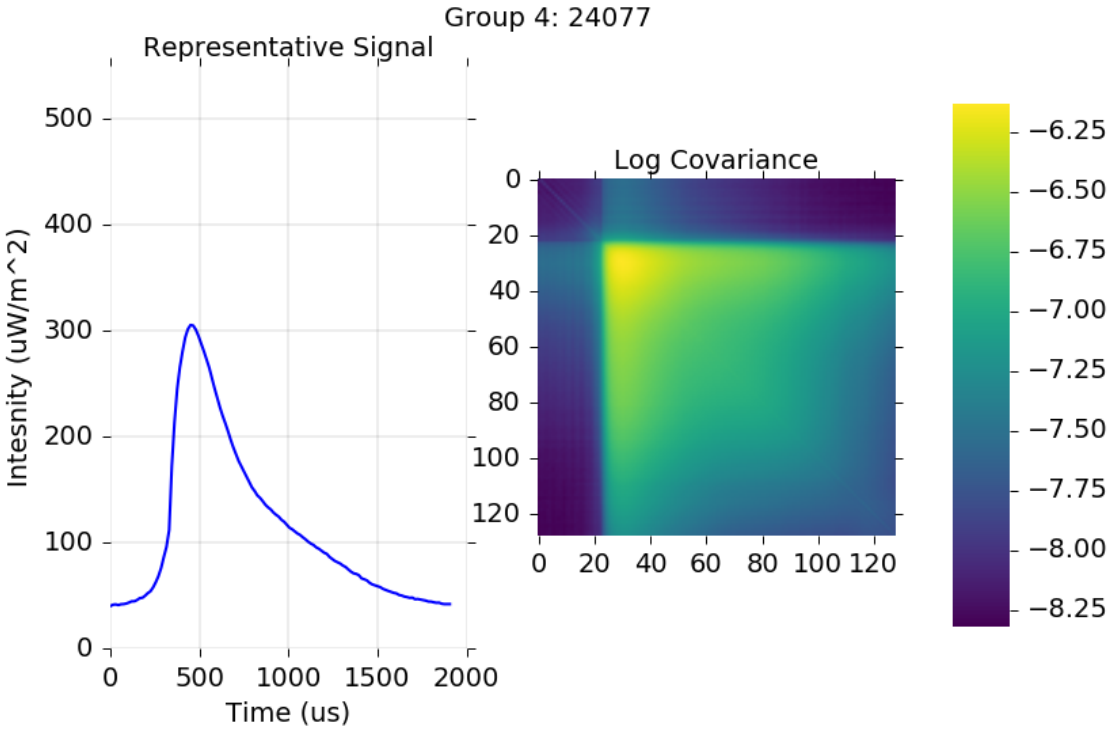

Figure A.4: Group 4 representative signal and covariance matrix.

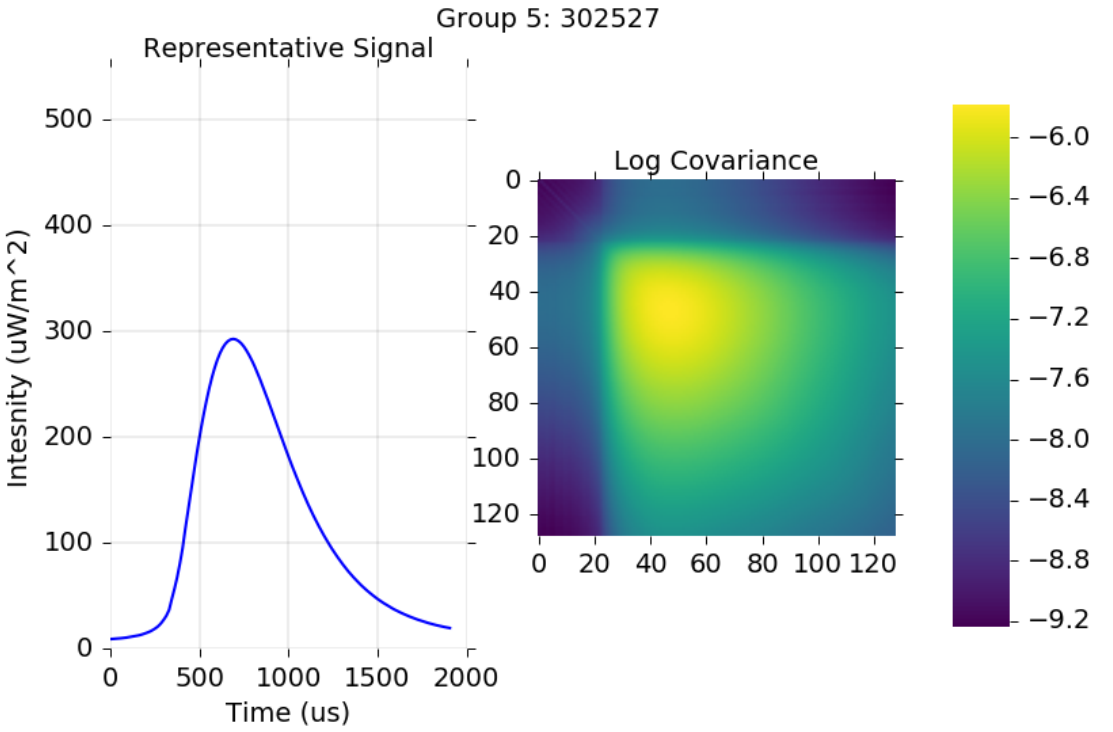

Figure A.5: Group 5 representative signal and covariance matrix.

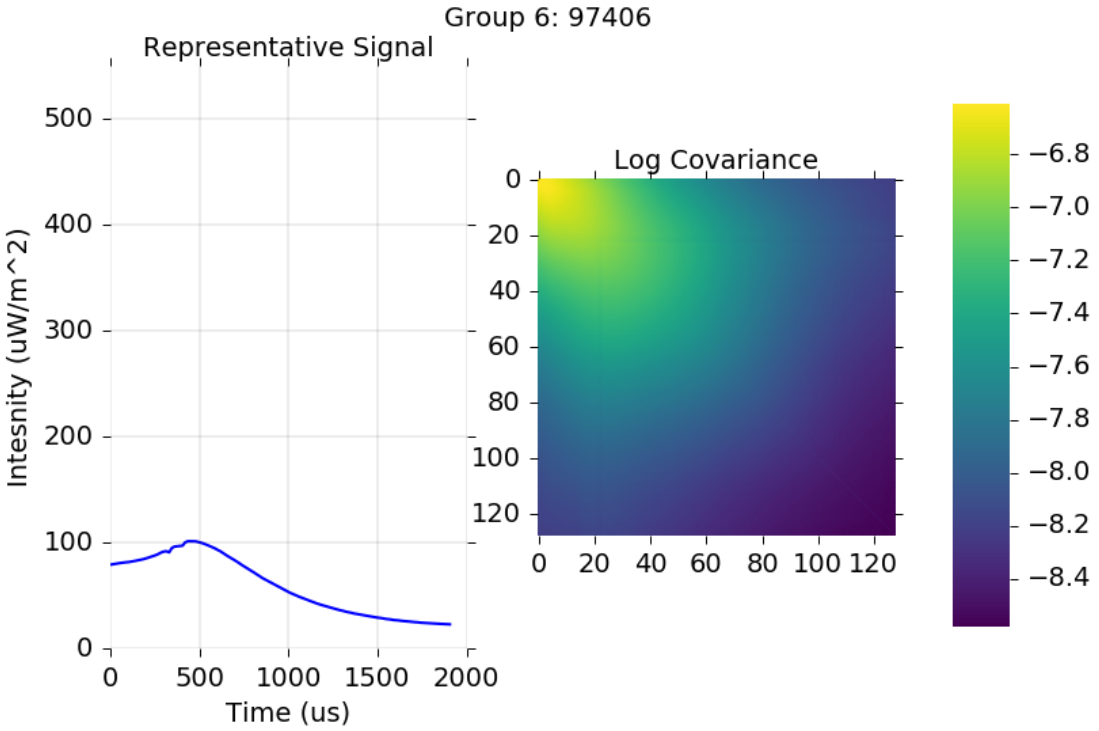

Figure A.6: Group 6 representative signal and covariance matrix.

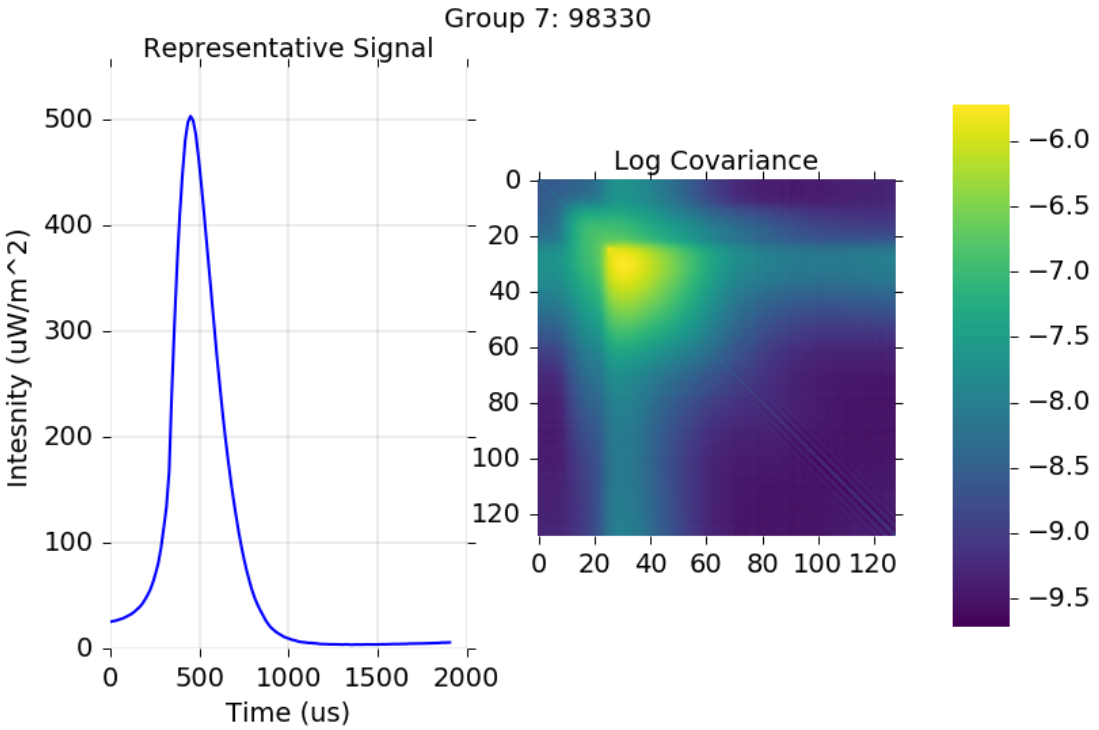

Figure A.7: Group 7 representative signal and covariance matrix.

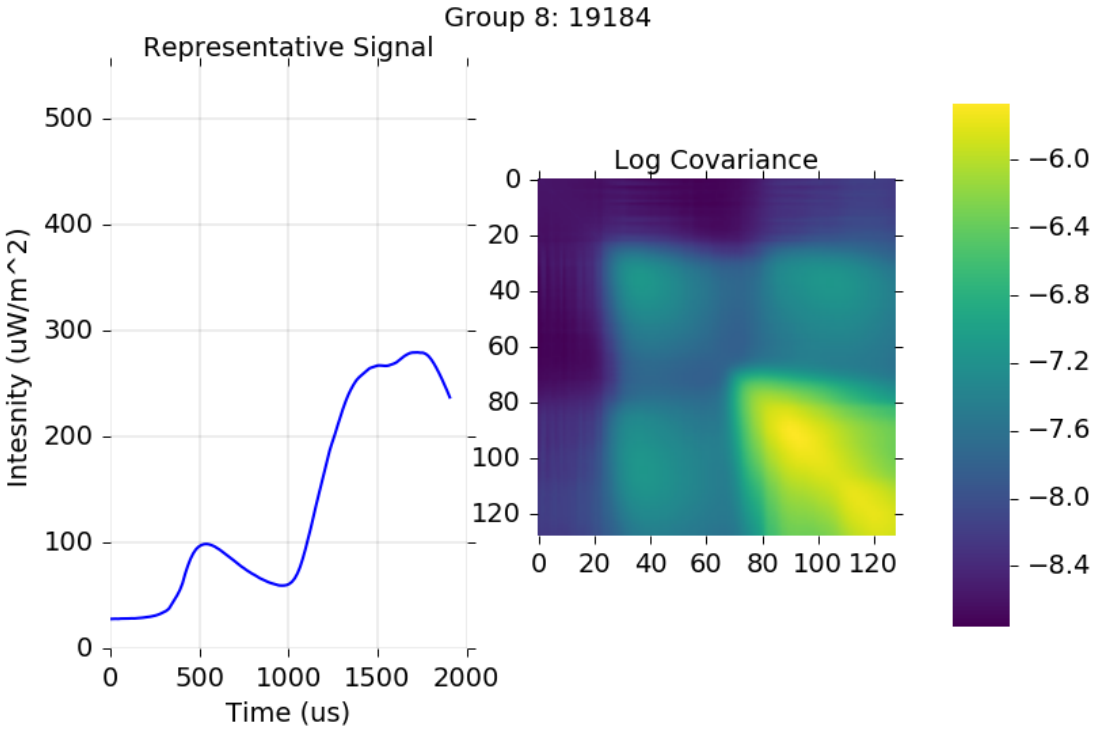

Figure A.8: Group 8 representative signal and covariance matrix.

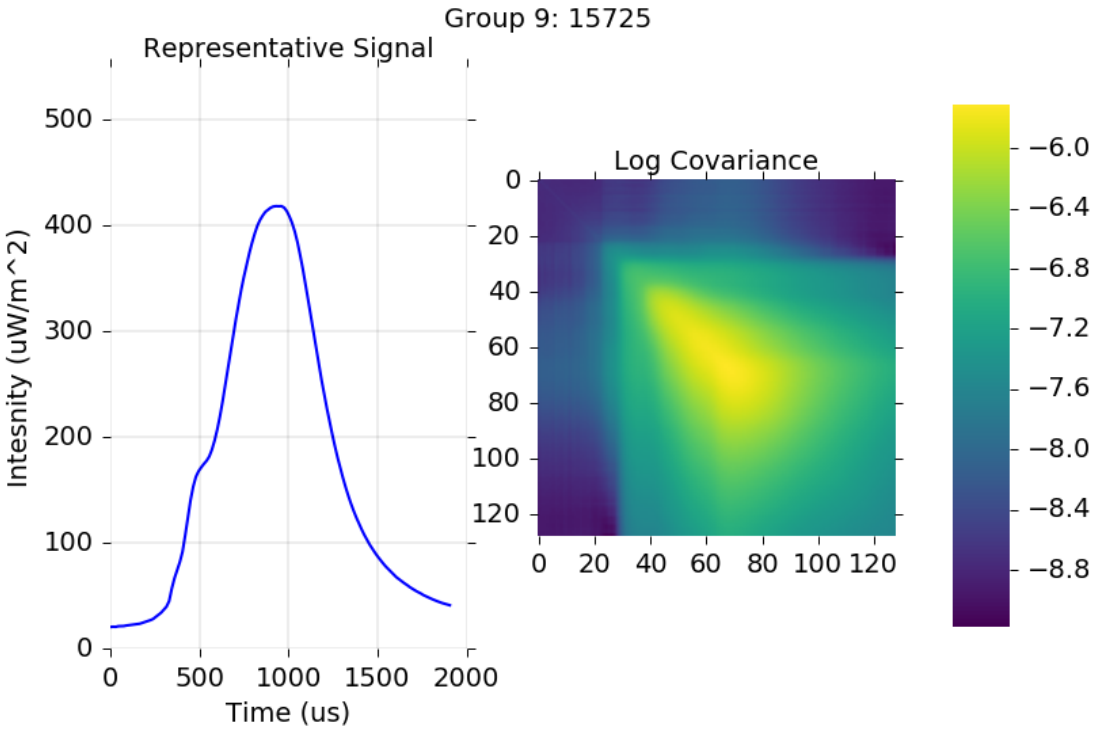

Figure A.9: Group 9 representative signal and covariance matrix.

APPENDIX B

DECILES

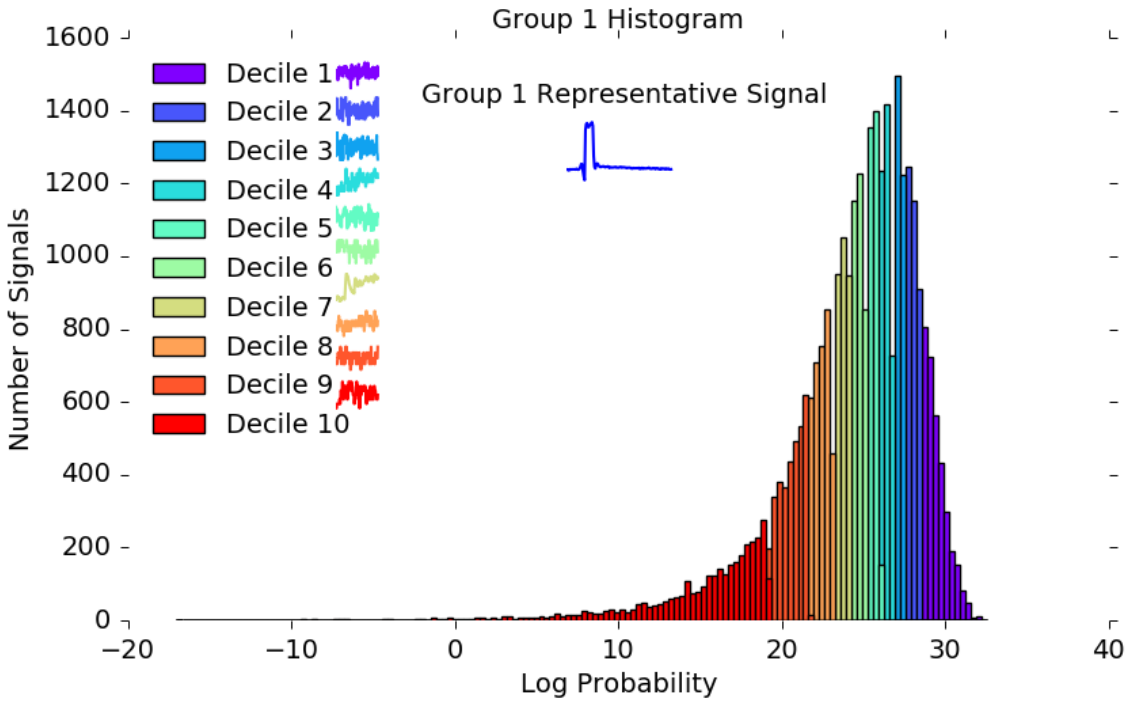

Figure B.1: Group 1 decile break down.

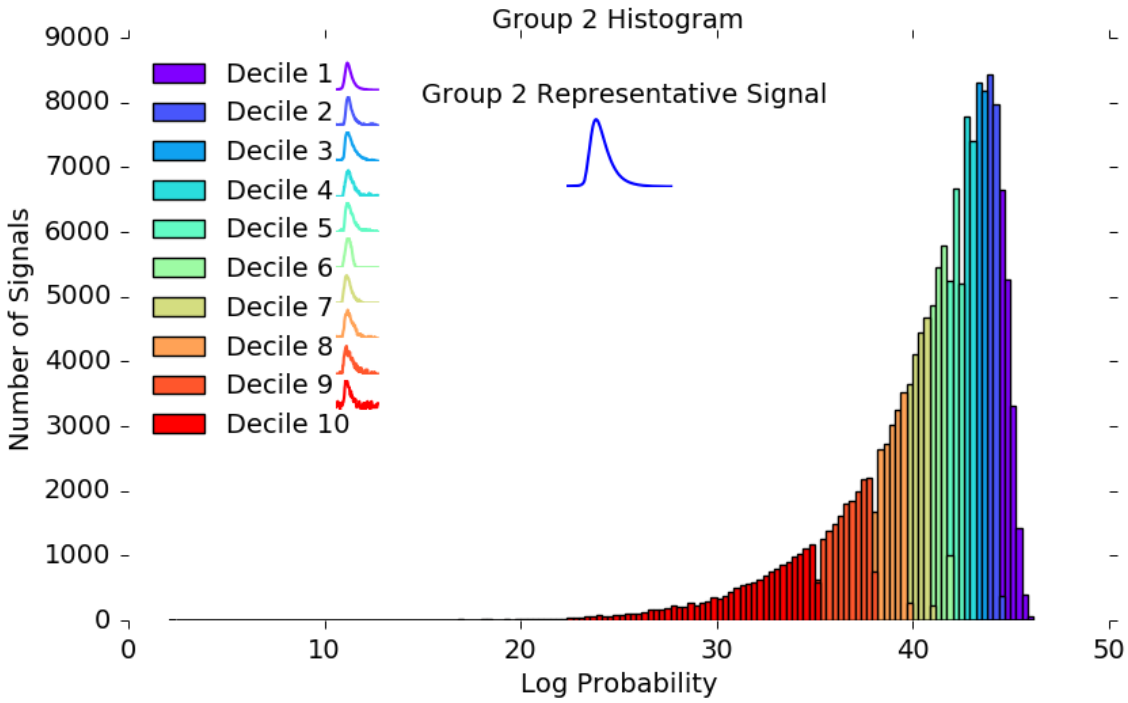

Figure B.2: Group 2 decile break down.

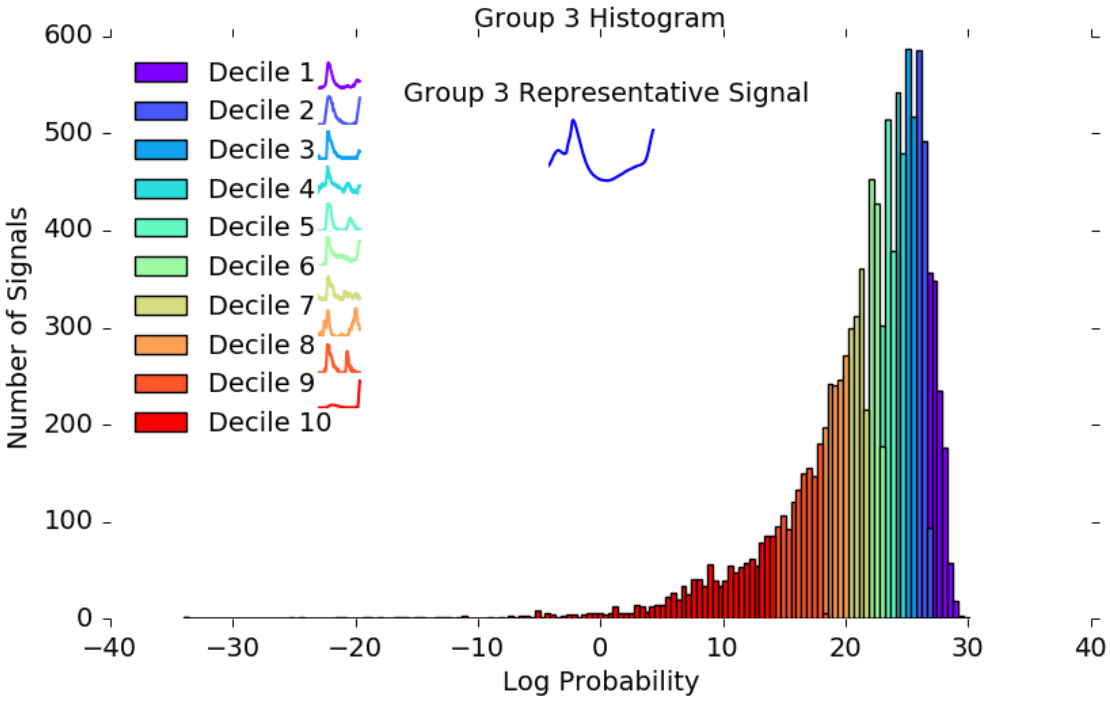

Figure B.3: Group 3 decile break down.

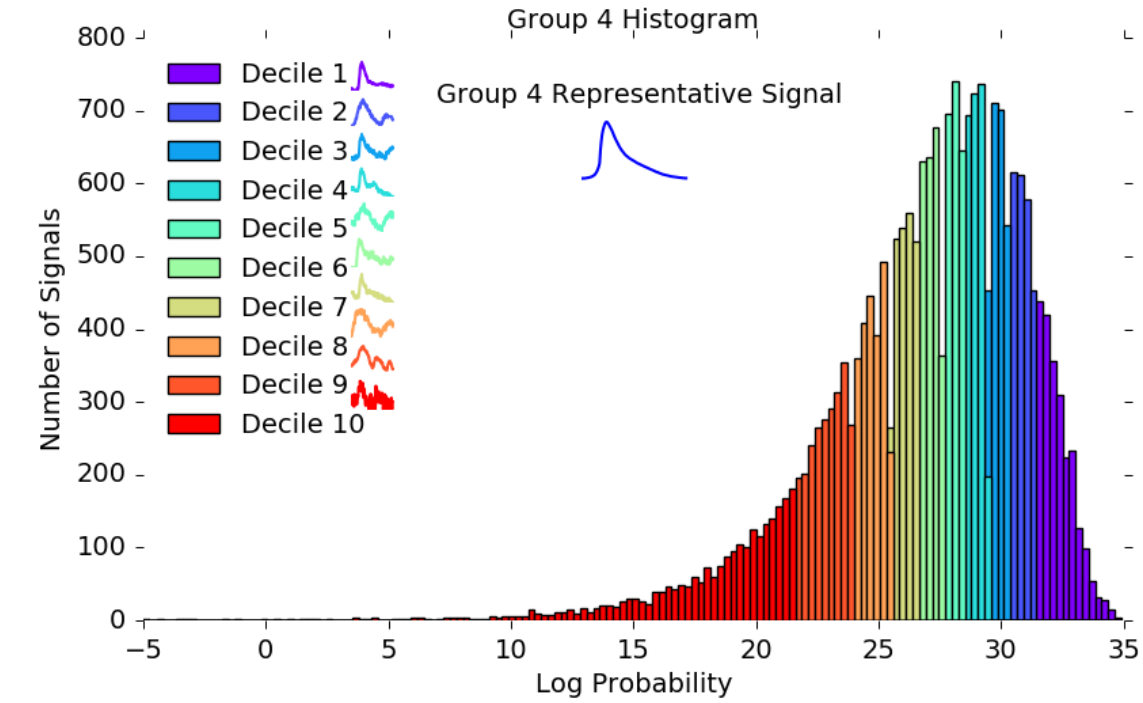

Figure B.4: Group 4 decile break down.

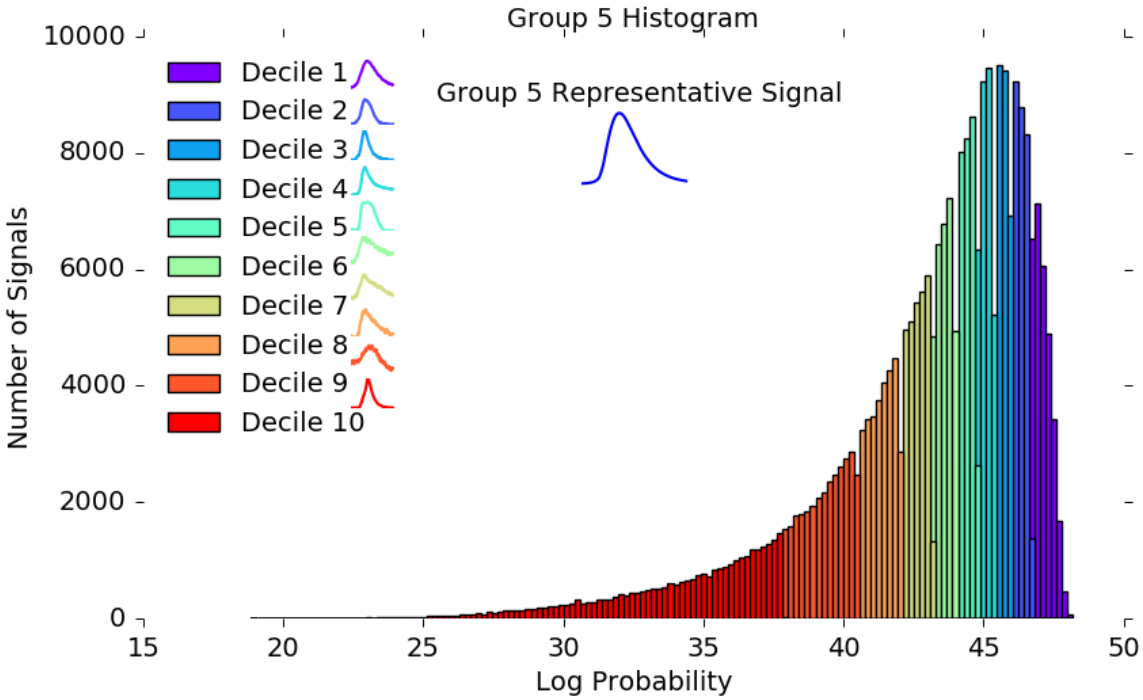

Figure B.5: Group 5 decile break down.

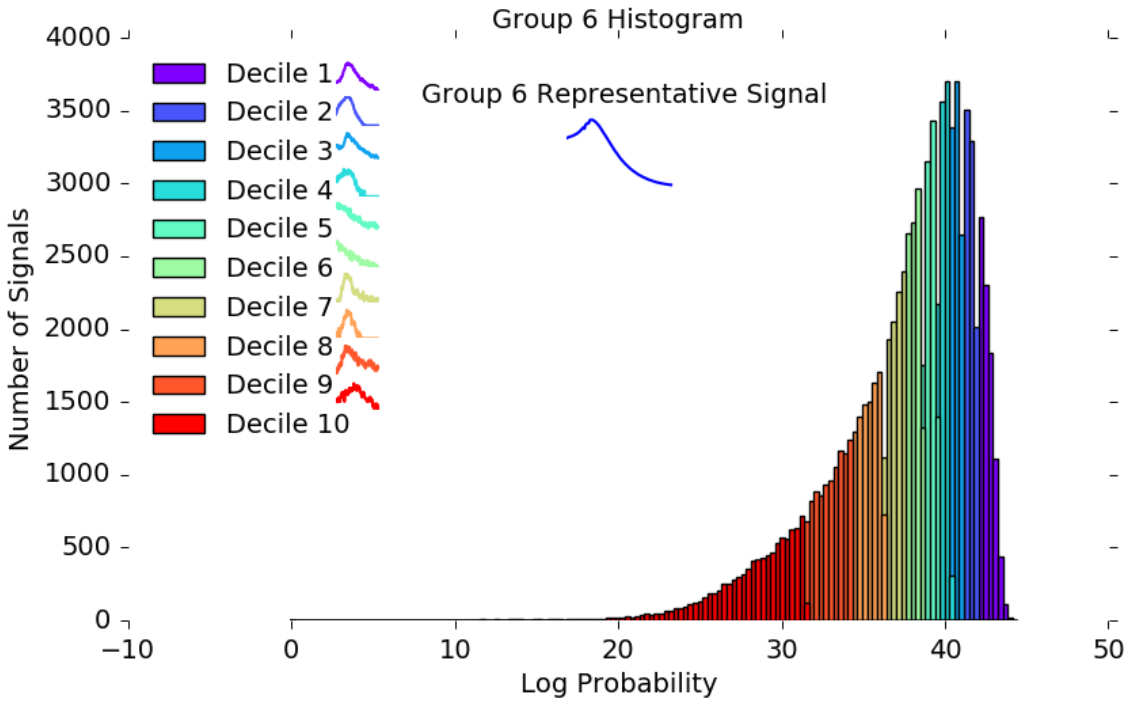

Figure B.6: Group 6 decile break down.

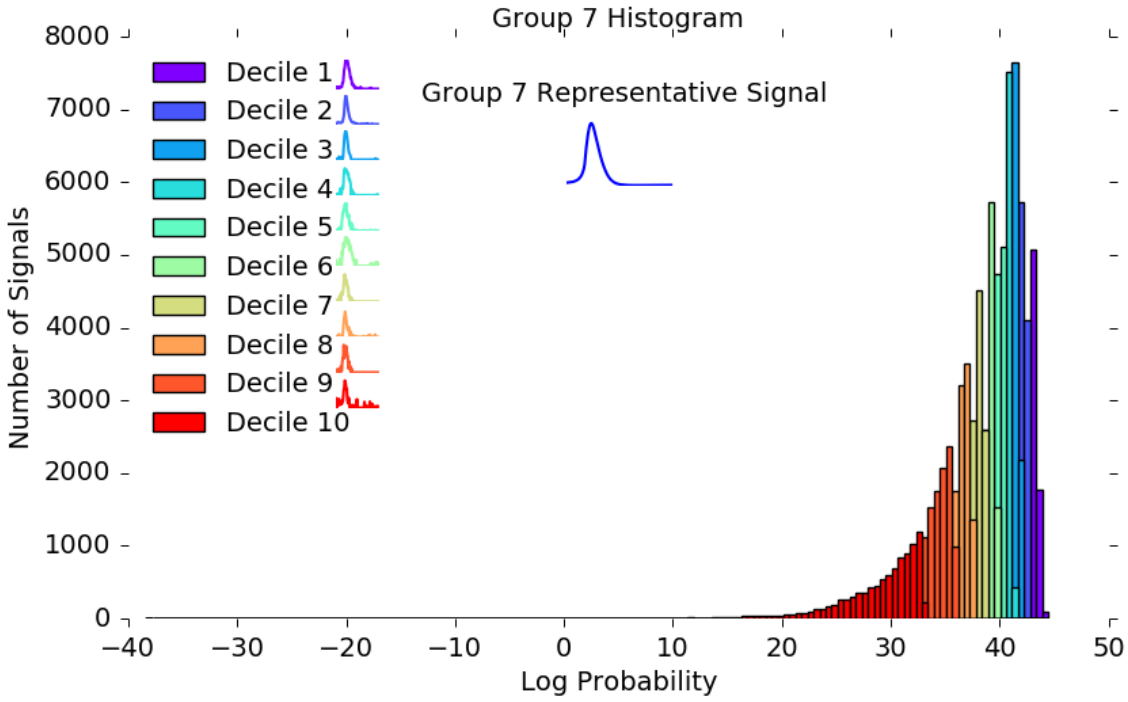

Figure B.7: Group 7 decile break down.

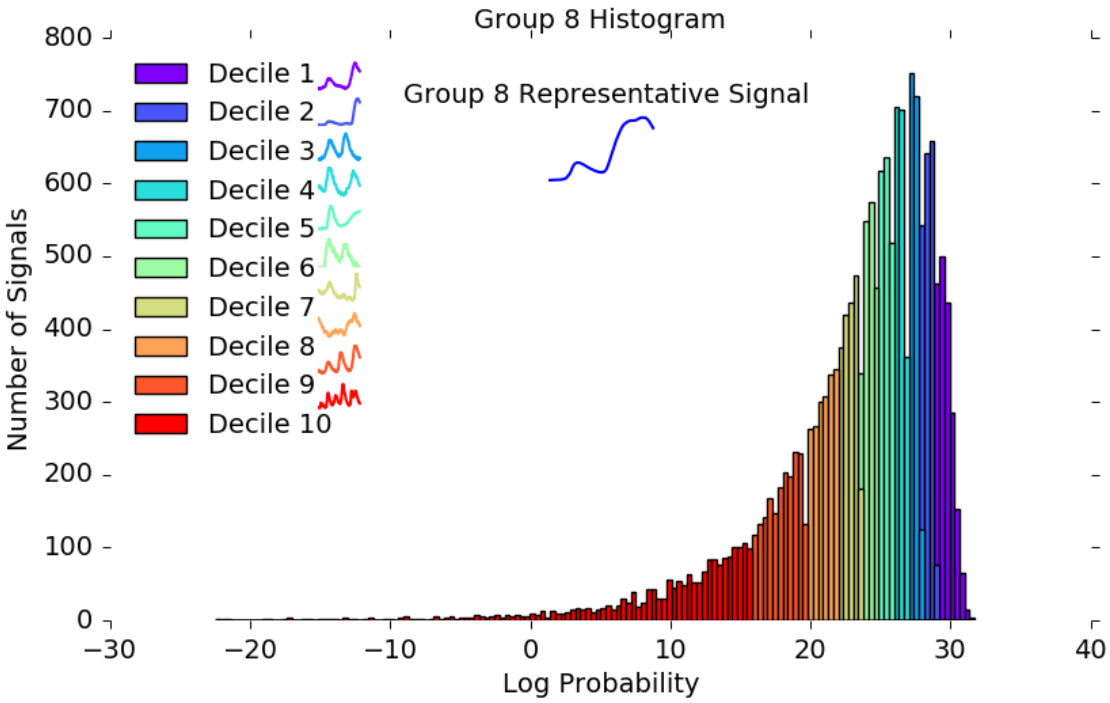

Figure B.8: Group 8 decile break down.

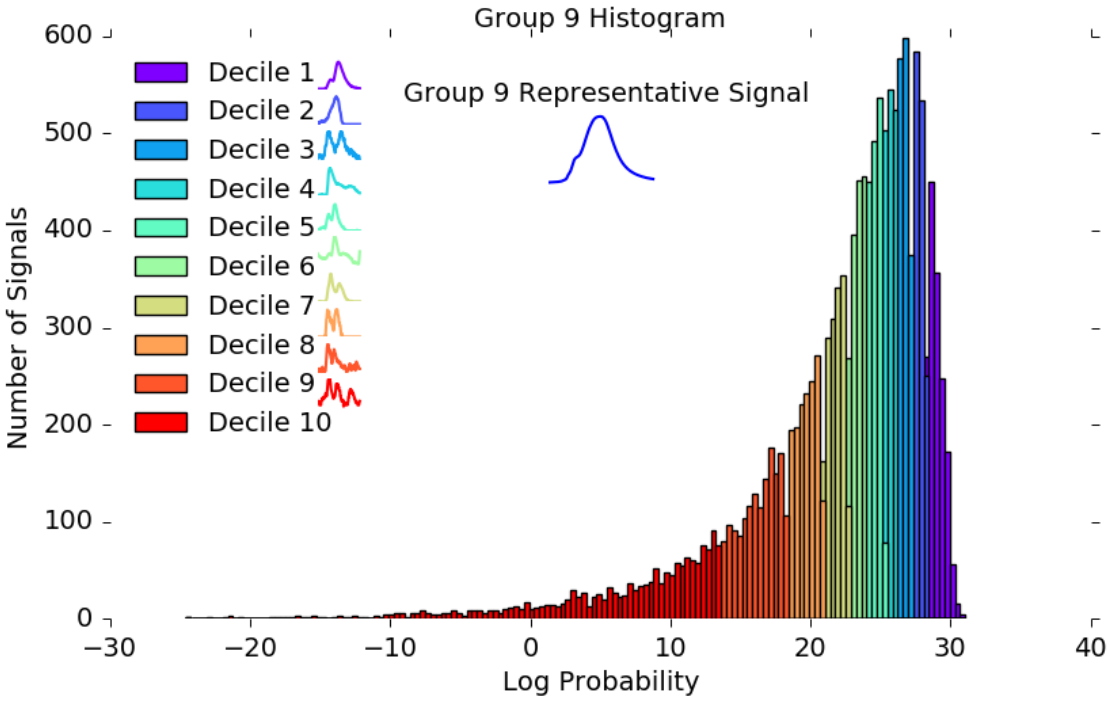

Figure B.9: Group 9 decile break down.

APPENDIX C

NOISE SYNTHESIS

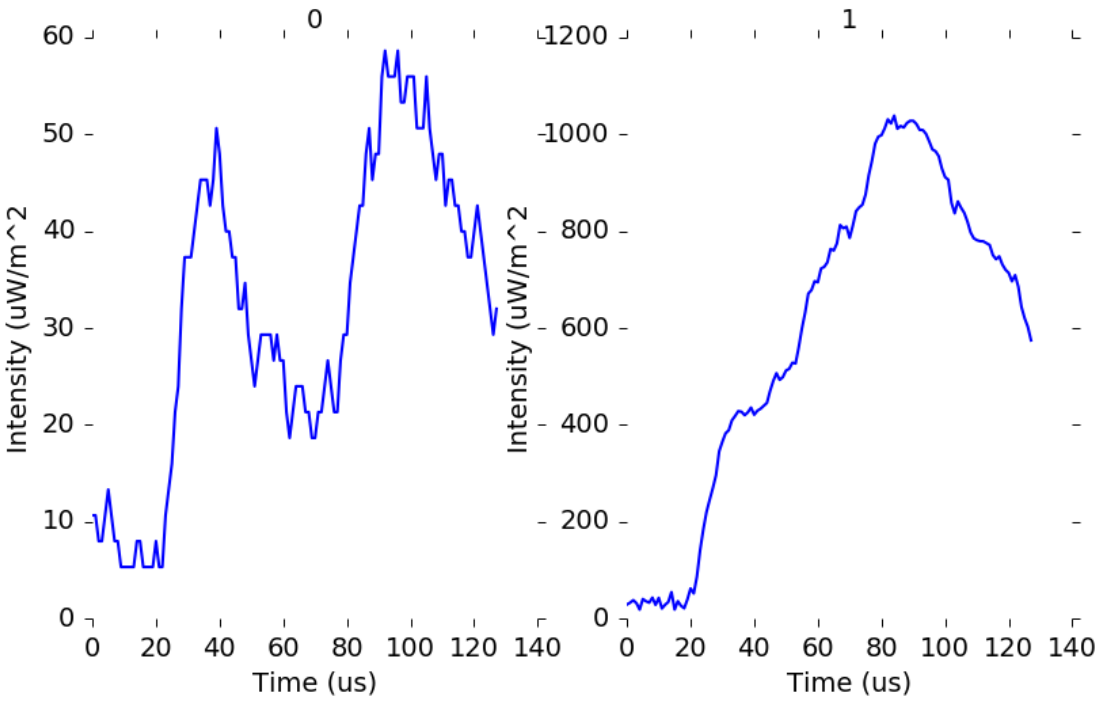

Figure C.1: Noise synthesis. 1 is synthesized.

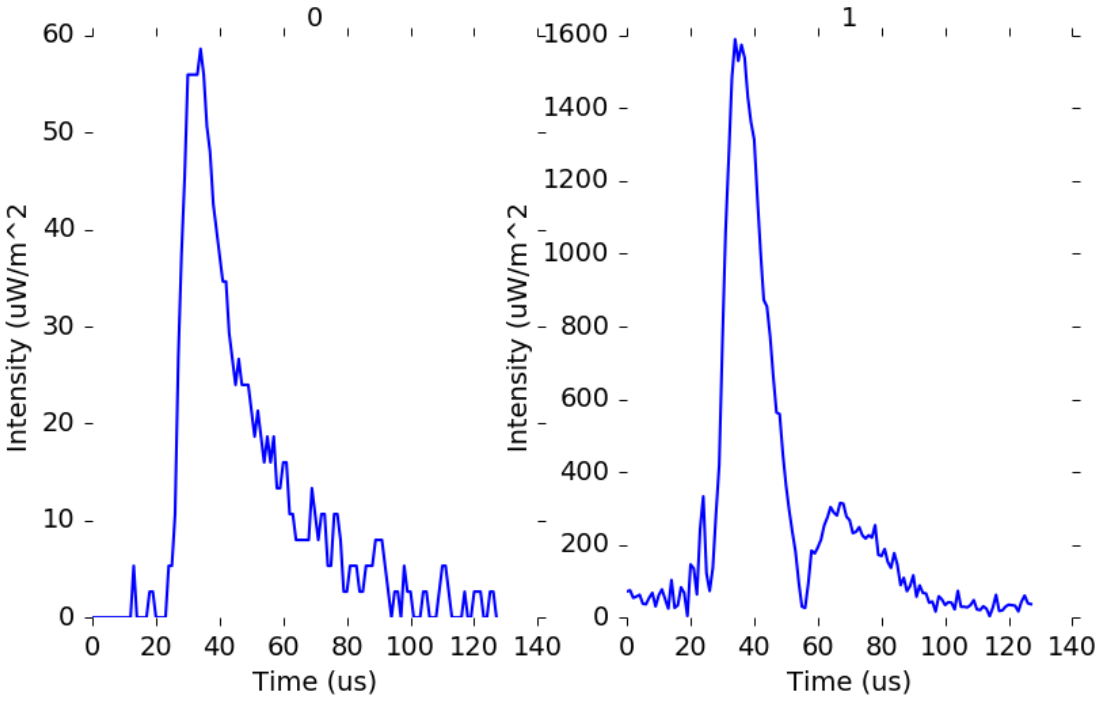

Figure C.2: Noise synthesis. 1 is synthesized.

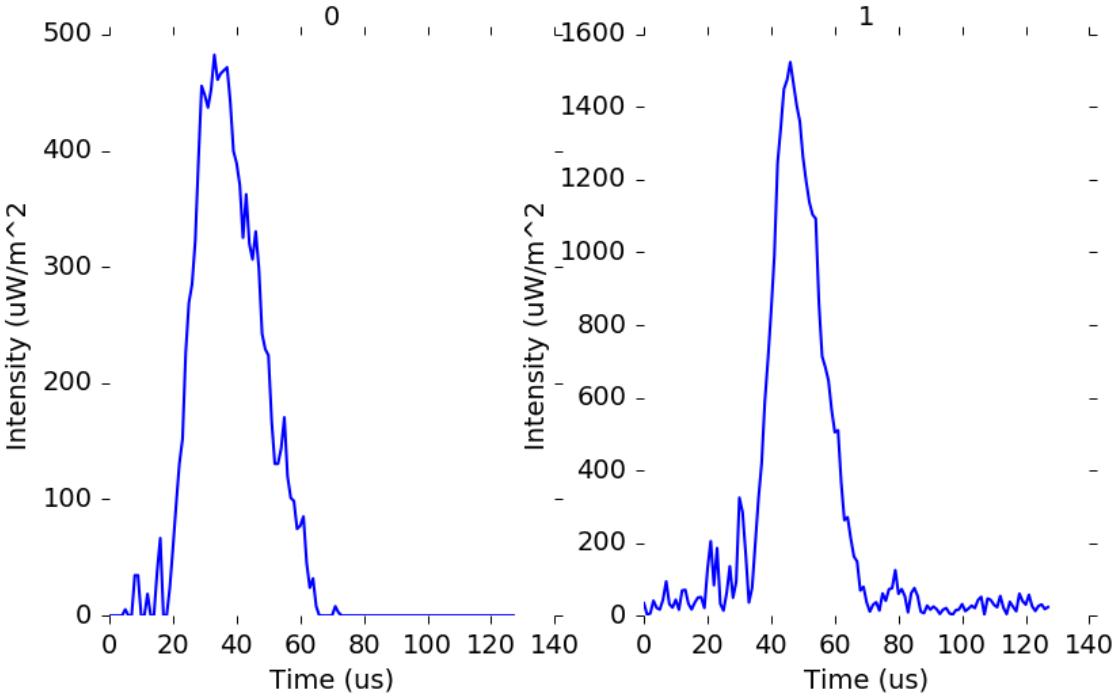

Figure C.3: Noise synthesis. 1 is synthesized.

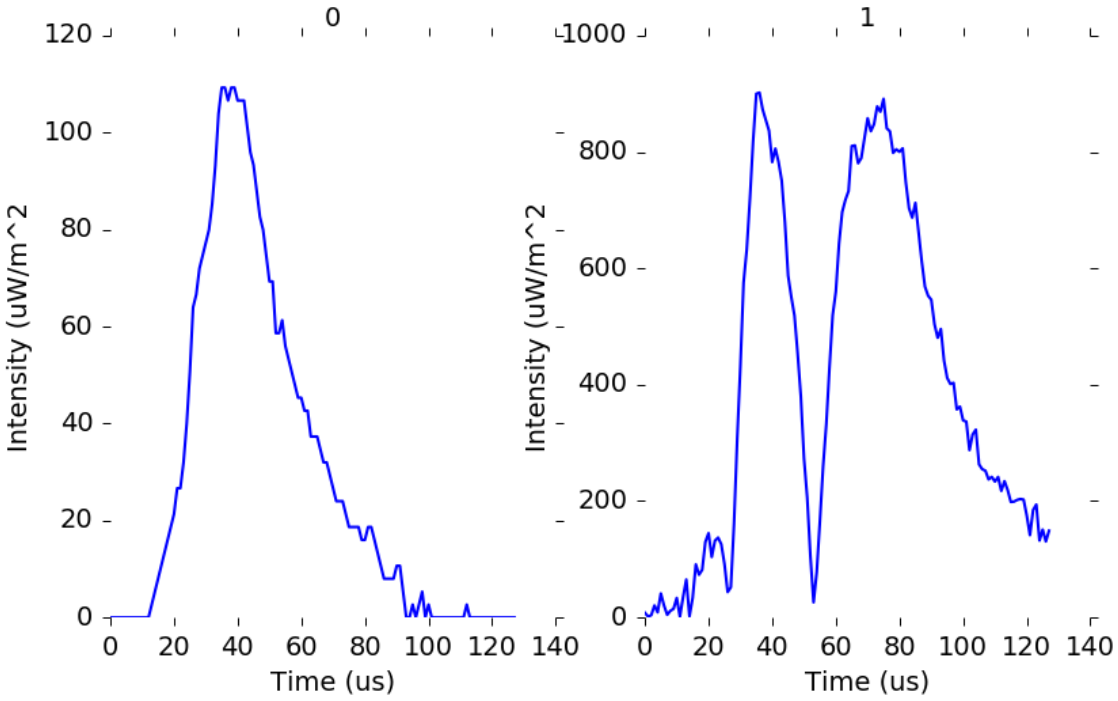

Figure C.4: Noise synthesis. 1 is synthesized.

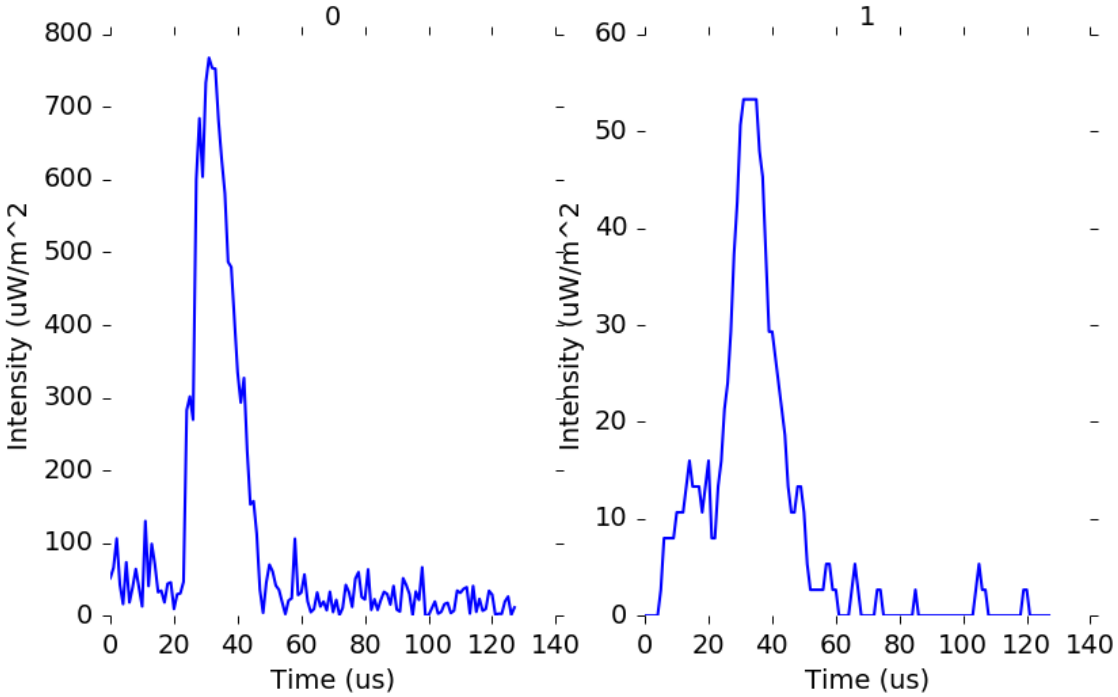

Figure C.5: Noise synthesis. 0 is synthesized.

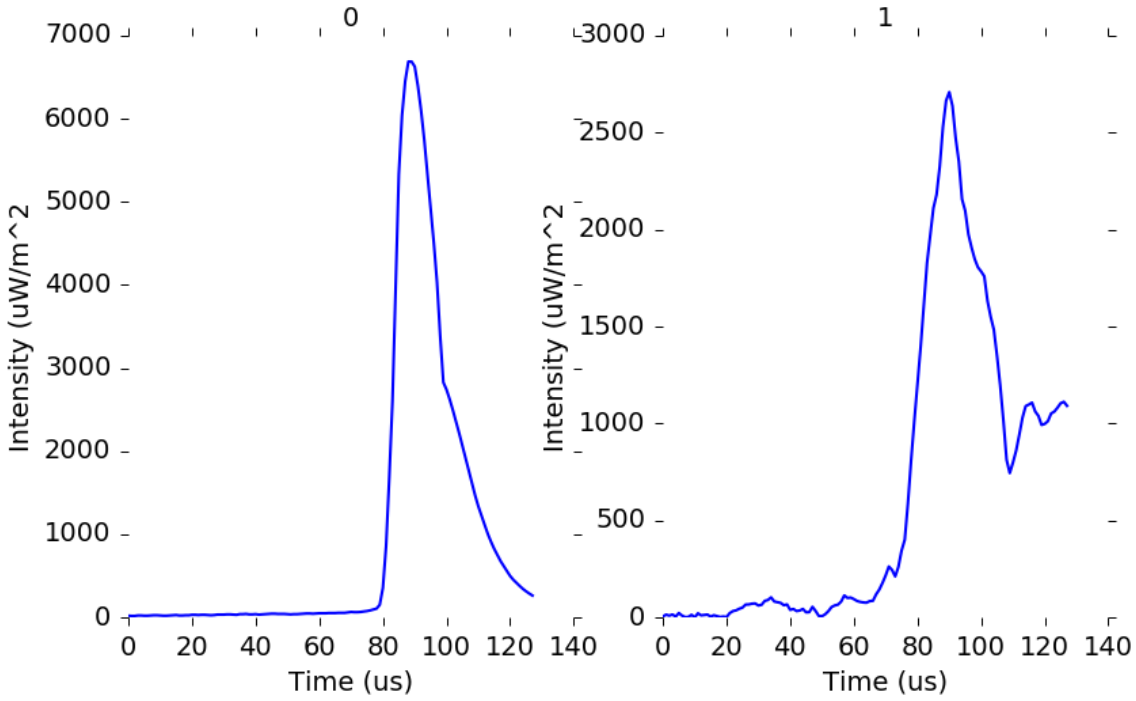

Figure C.6: Noise synthesis. 1 is synthesized.

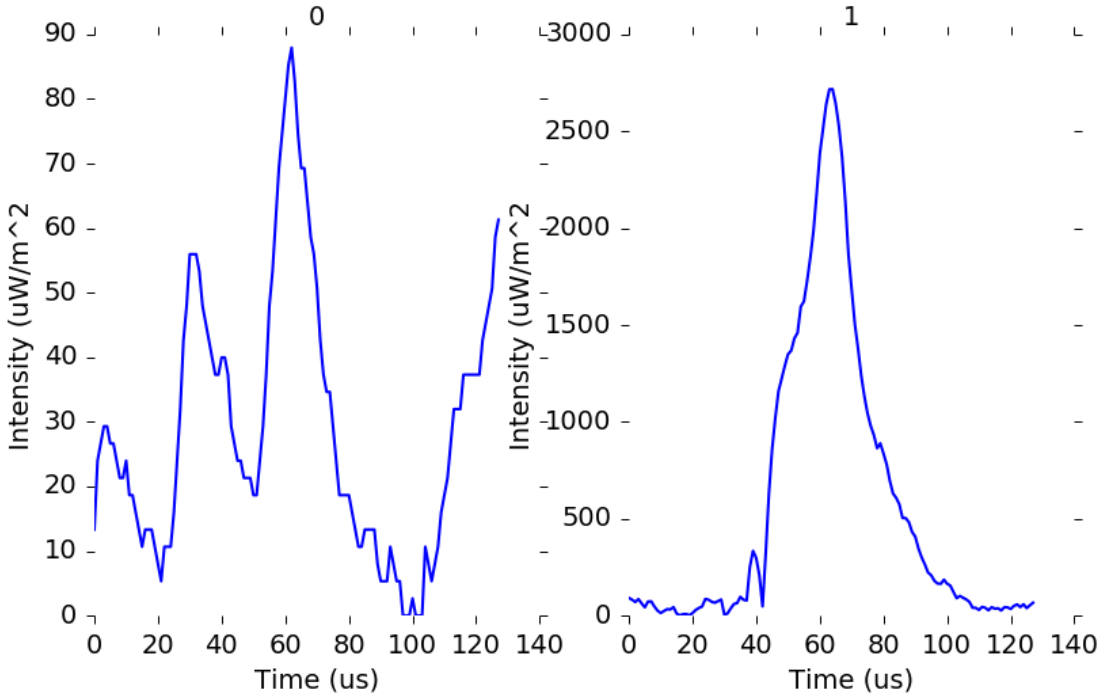

Figure C.7: Noise synthesis. 1 is synthesized.

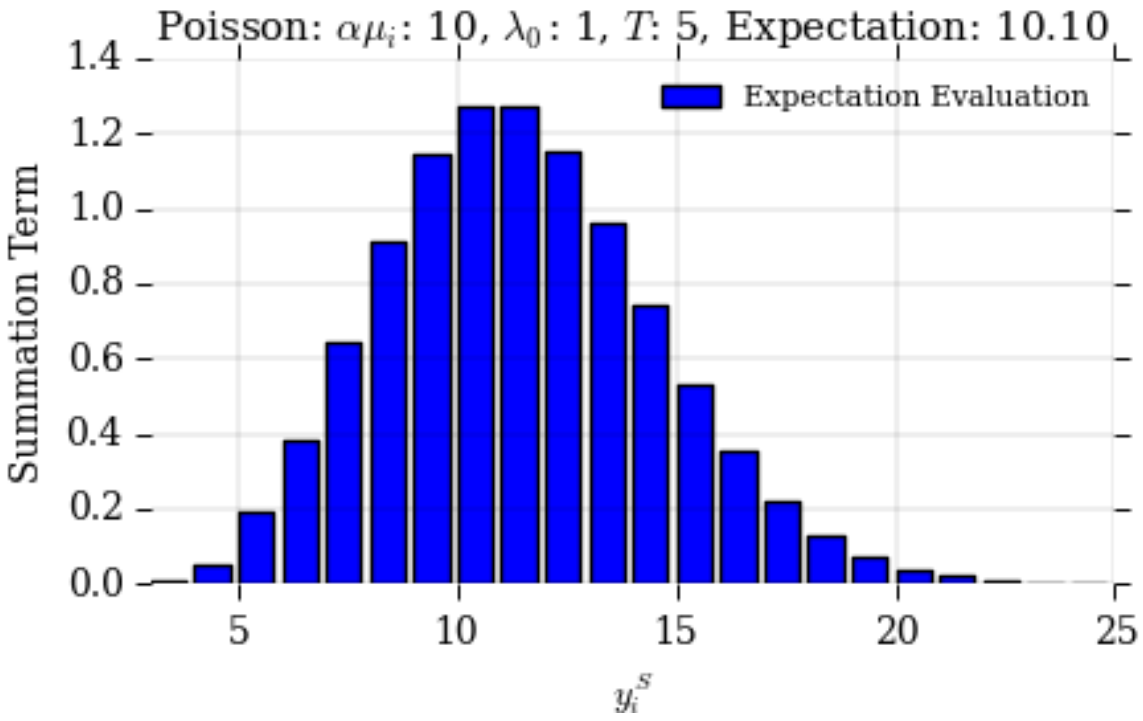

Figure D.1: Low noise, entirely saturated, low photon count.

APPENDIX D

TERMS OF CONDITIONAL EXPECTATIONS – POISSON

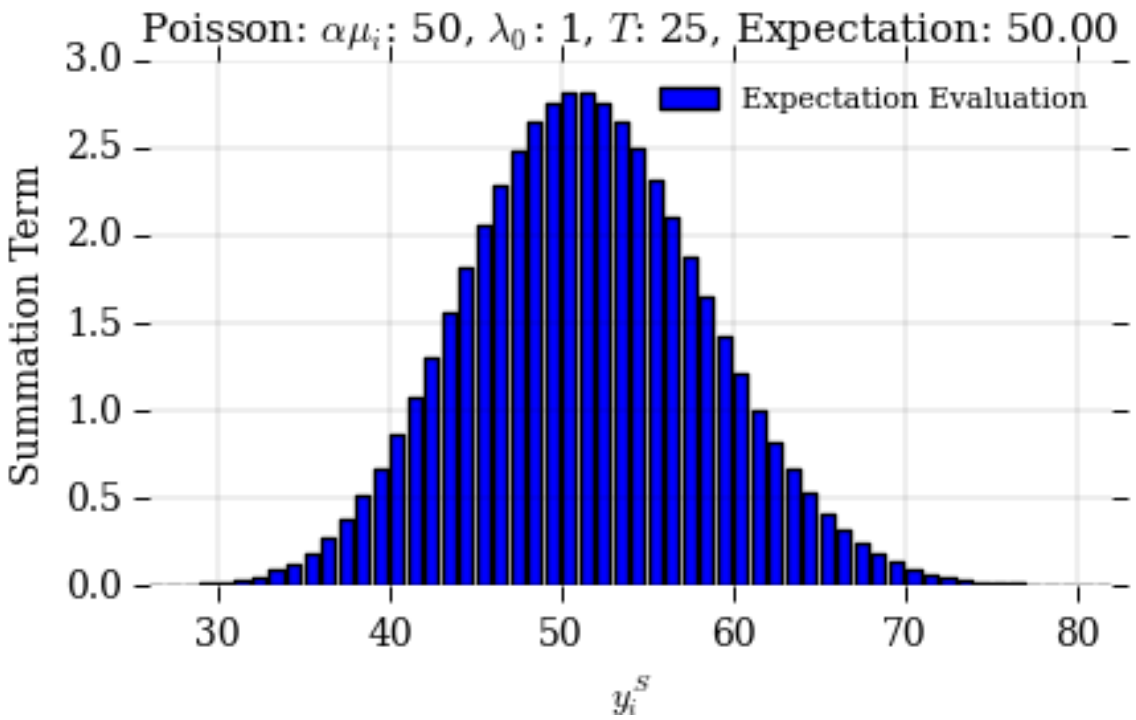

Figure D.2: Low noise, entirely saturated, high photon count.

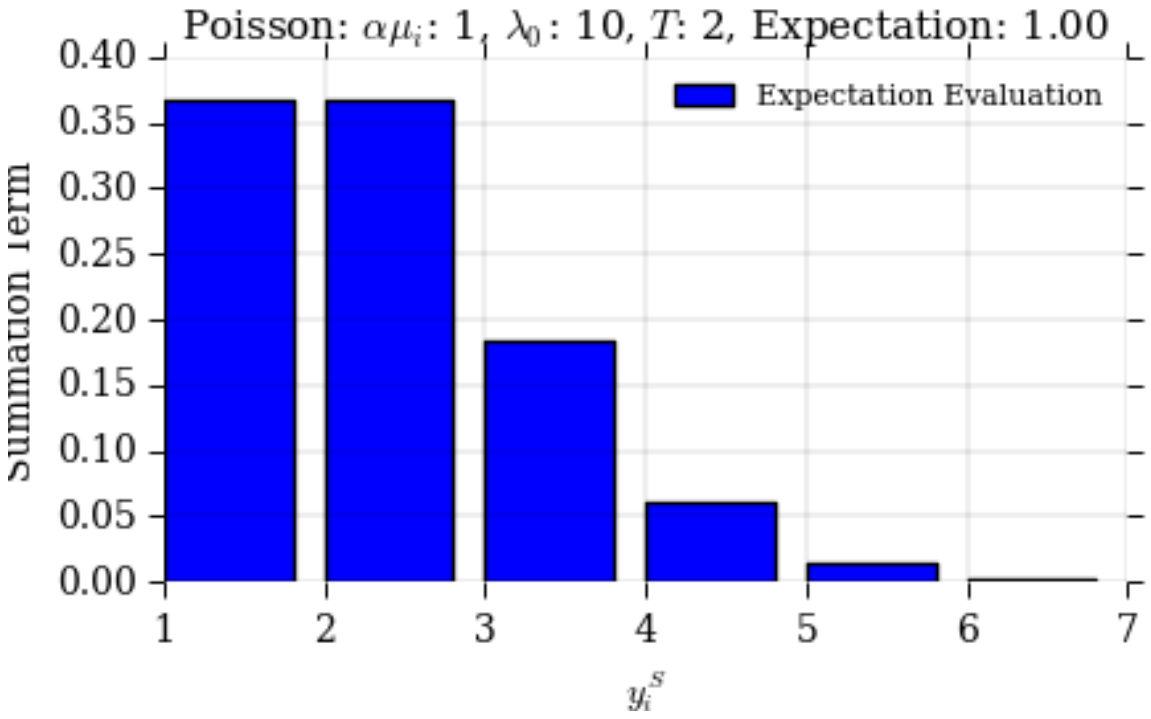

Figure D.3: High noise, entirely saturated, low photon count.

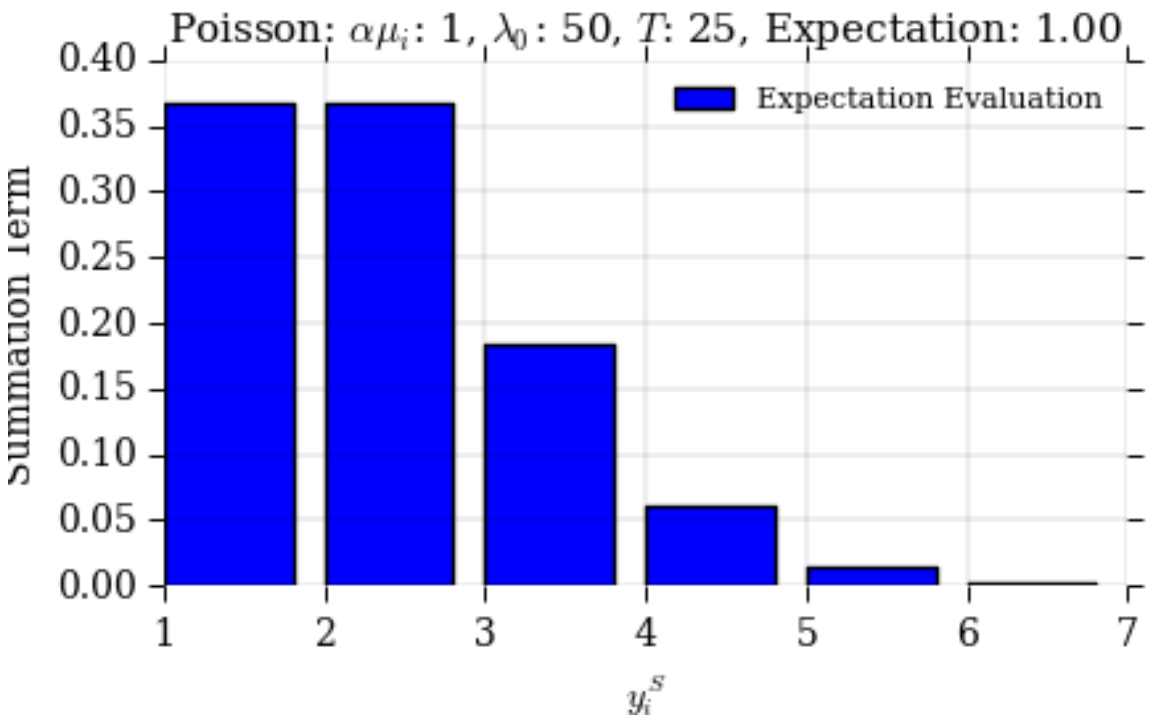

Figure D.4: High noise, entirely saturated, high photon count.

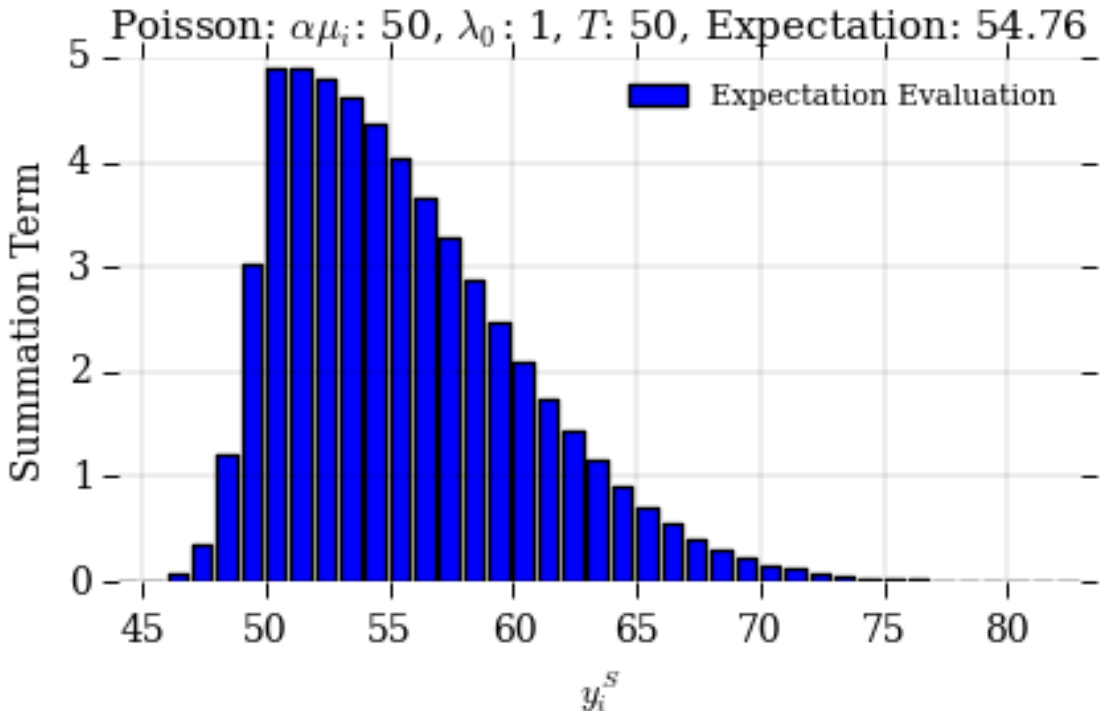

Figure D.5: Low noise, partially saturated, high photon count.

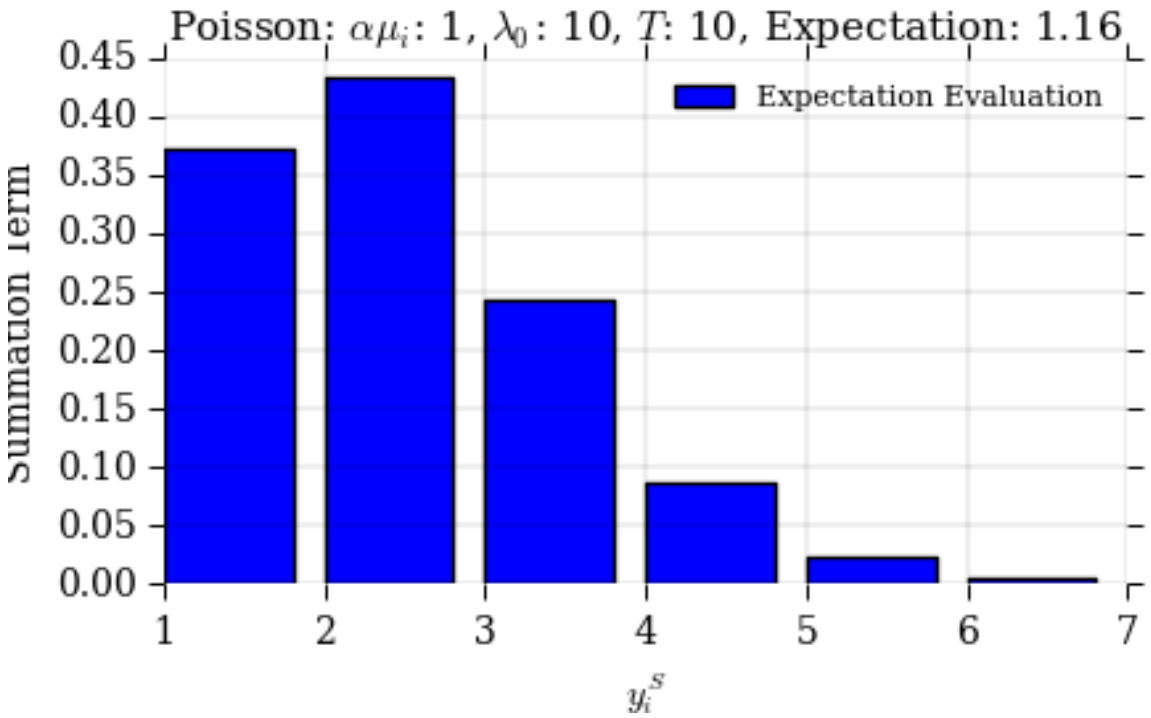

Figure D.6: High noise, partly saturated, low photon count.

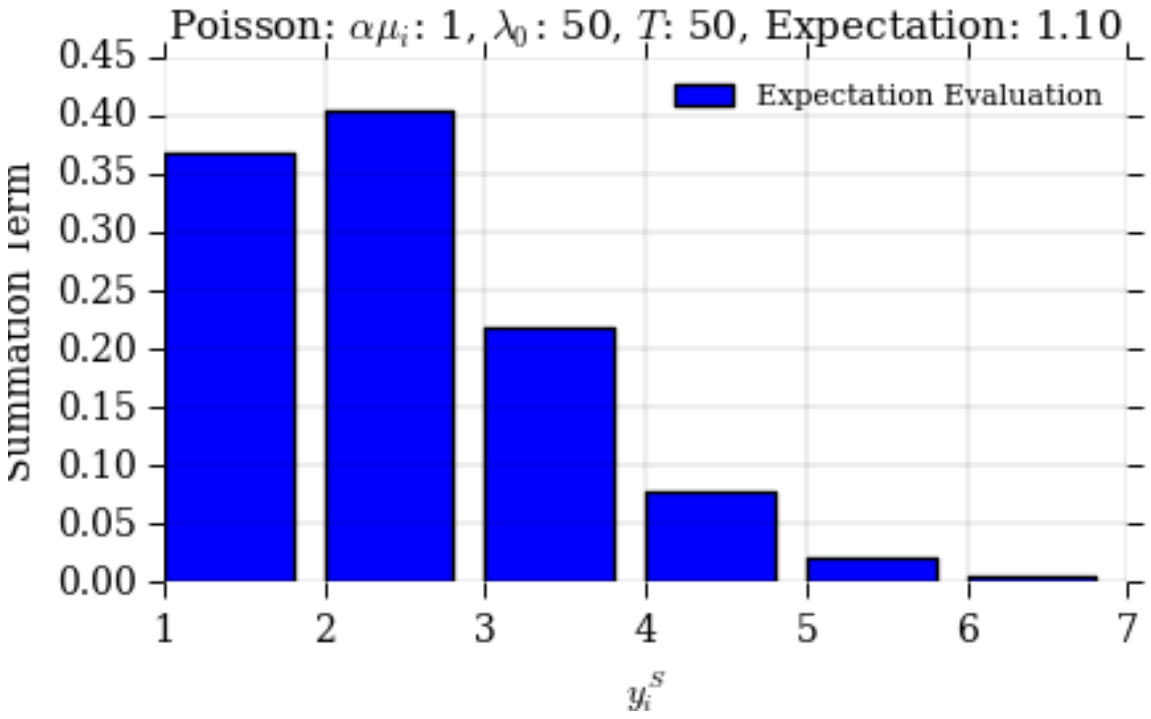

Figure D.7: High noise, partially saturated, high photon count.

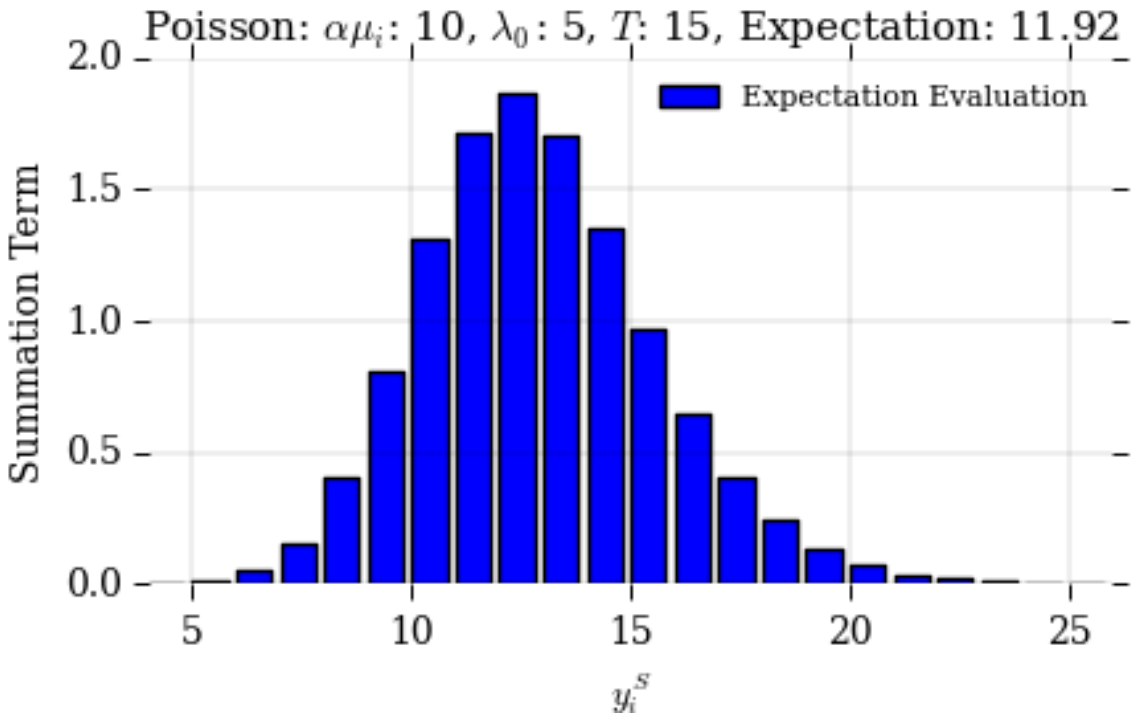

Figure D.8: Moderate noise, partially saturated, low photon count.

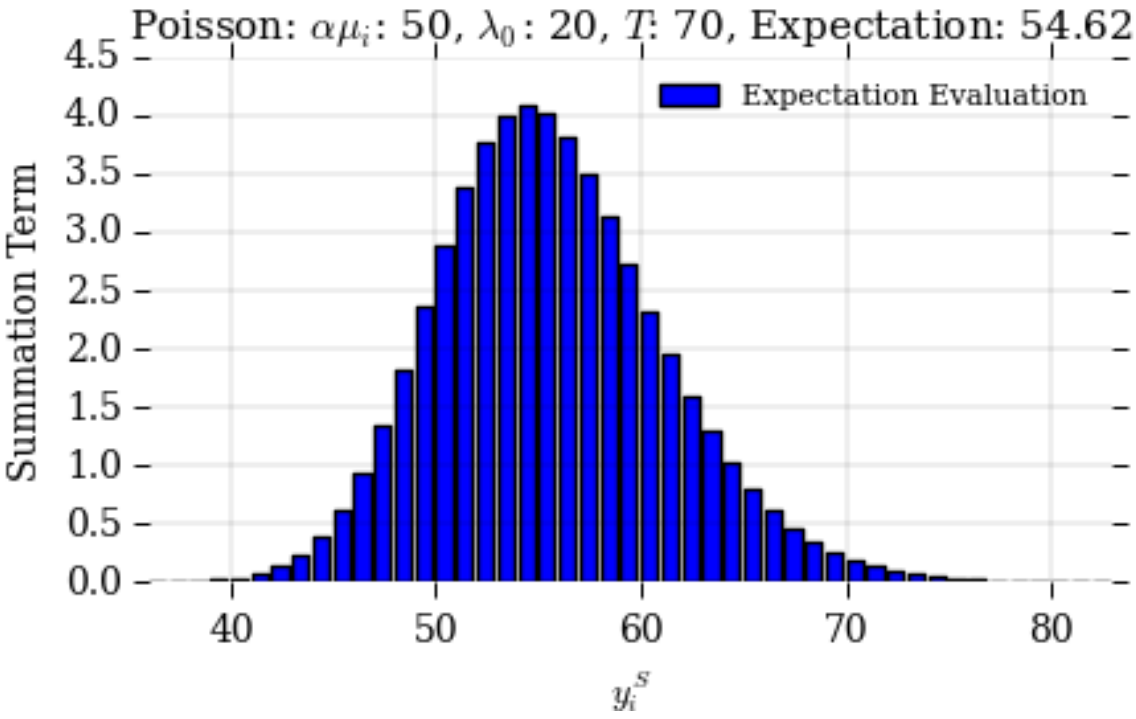

Figure D.9: Moderate noise, partially saturated, high photon count.

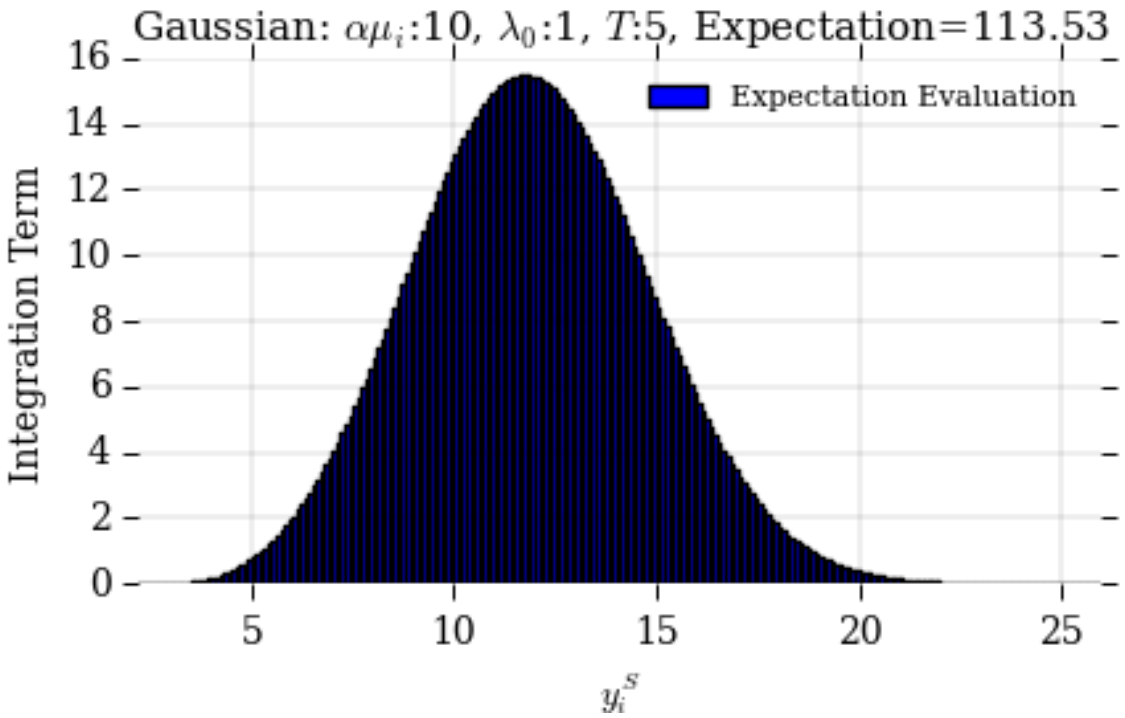

Figure E.1: Low noise, entirely saturated, low photon count.

APPENDIX E

INTEGRANDS OF CONDITIONAL EXPECTATIONS – GAUSSIAN

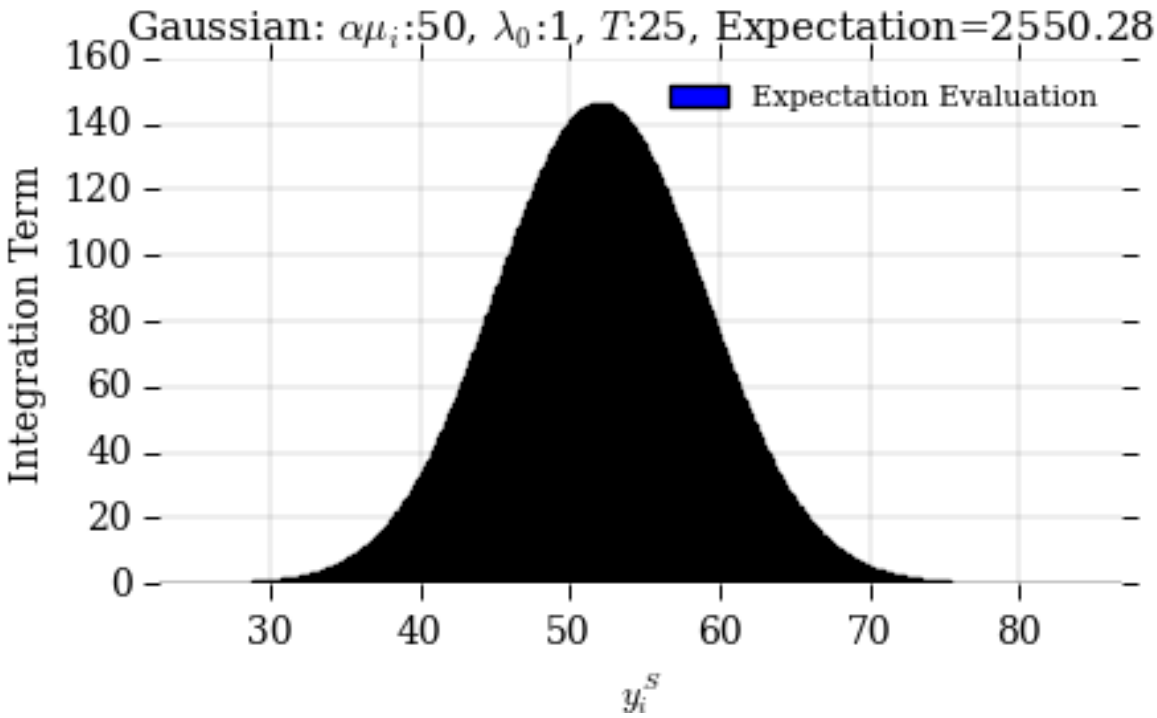

Figure E.2: Low noise, entirely saturated, high photon count.

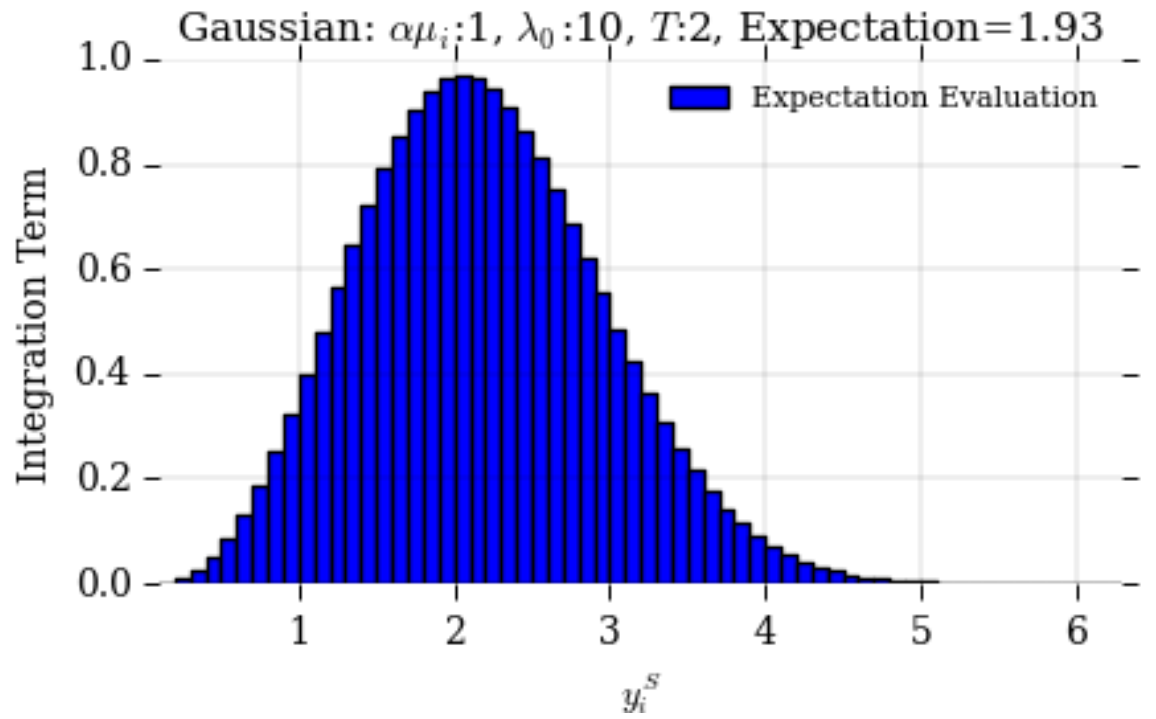

Figure E.3: High noise, entirely saturated, low photon count.

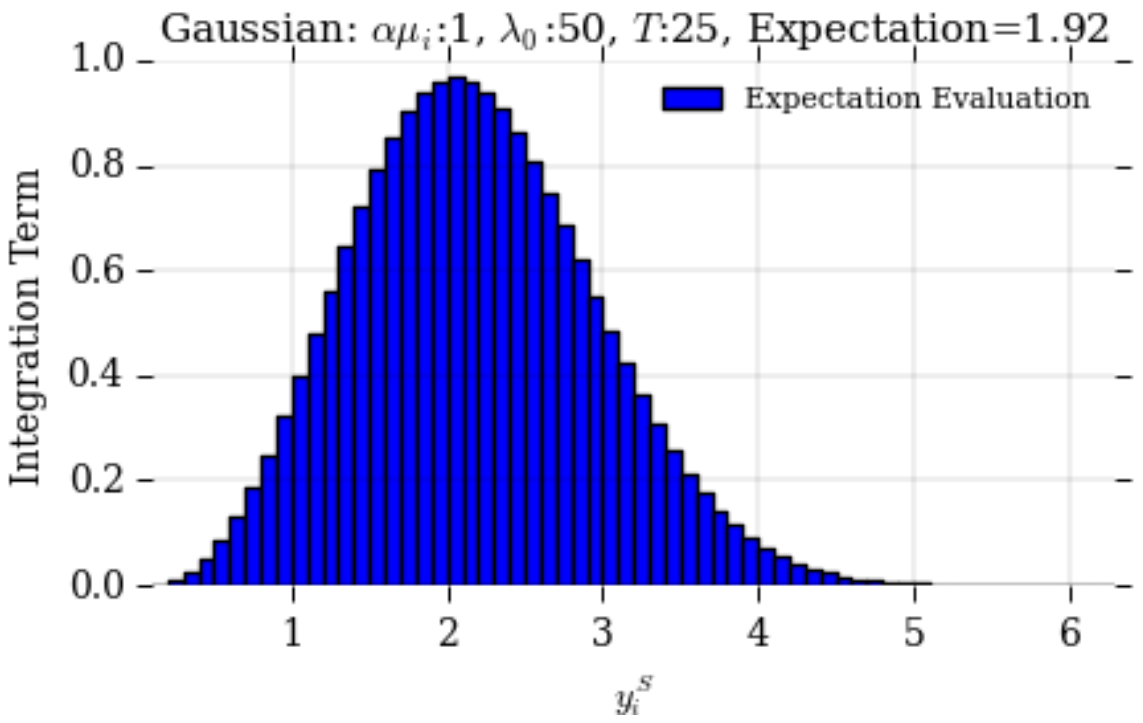

Figure E.4: High noise, entirely saturated, high photon count.

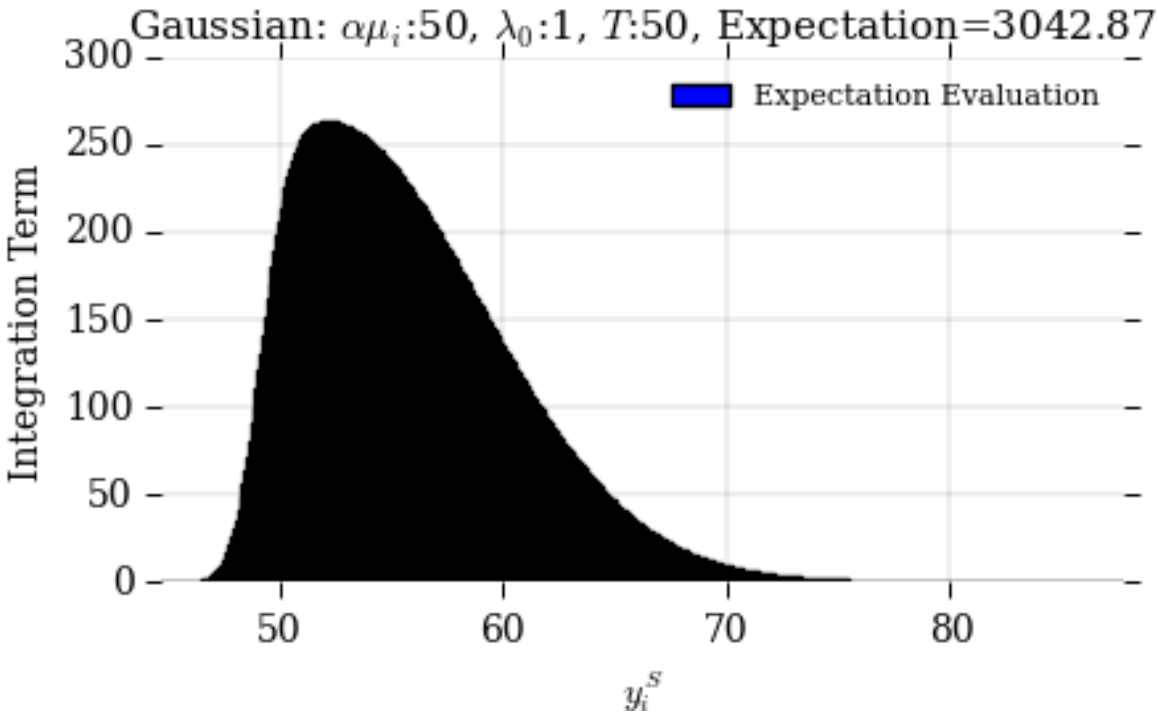

Figure E.5: Low noise, partially saturated, high photon count.

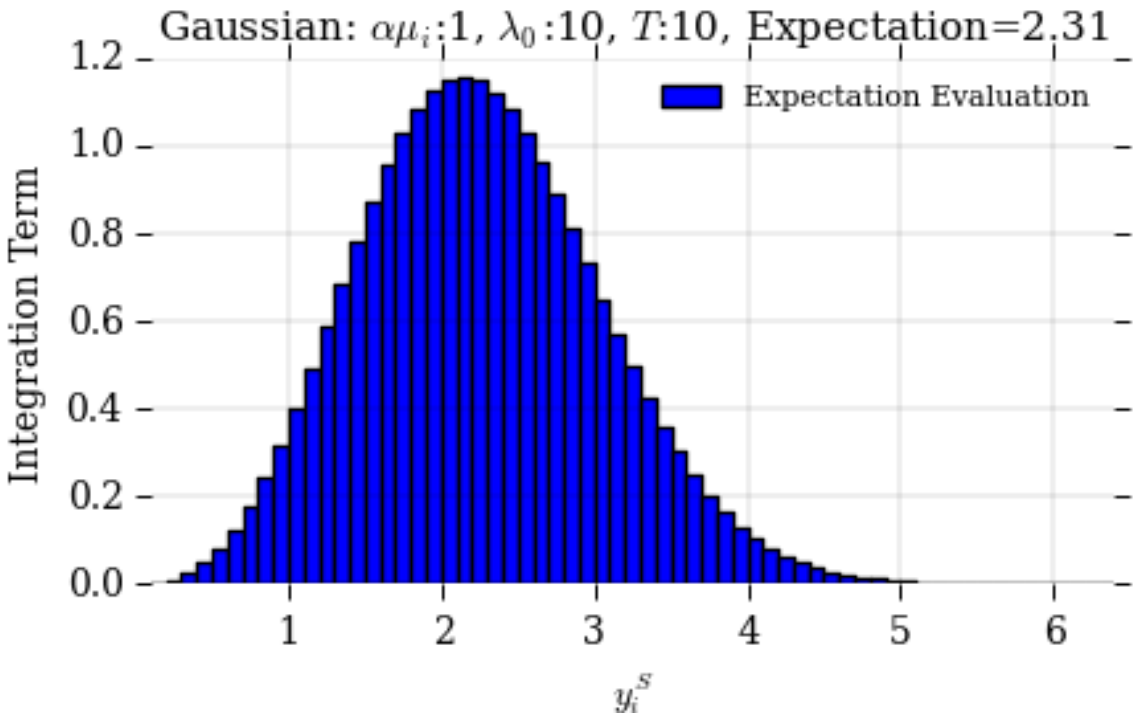

Figure E.6: High noise, partly saturated, low photon count.

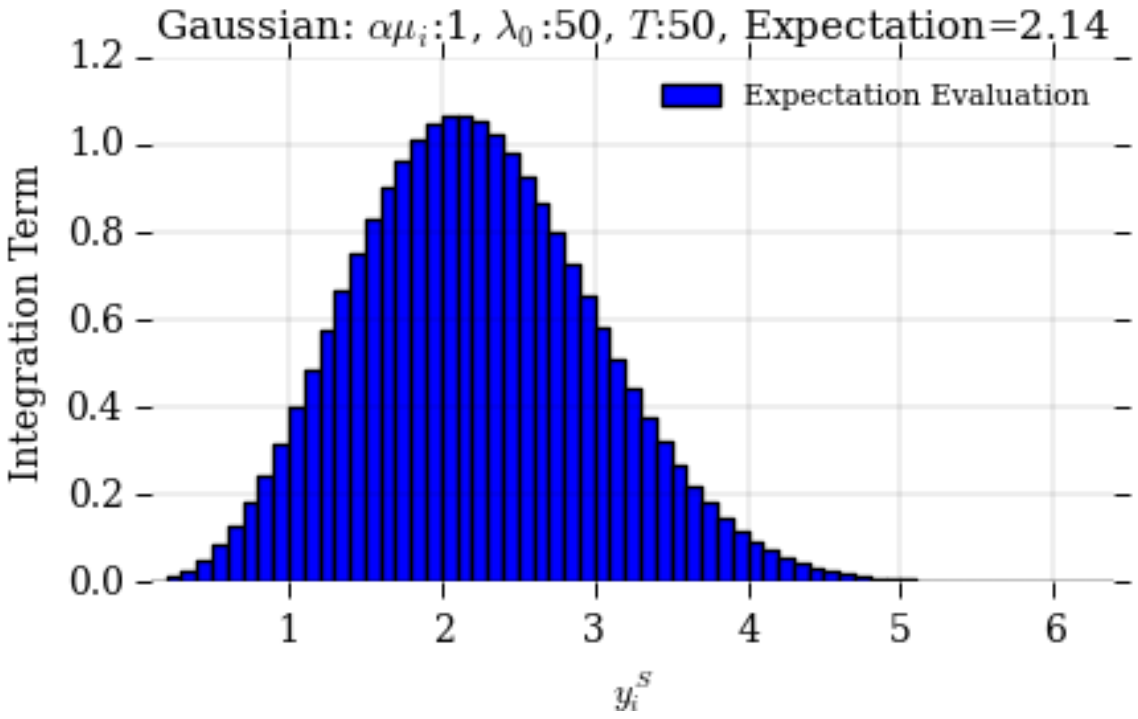

Figure E.7: High noise, partially saturated, high photon count.

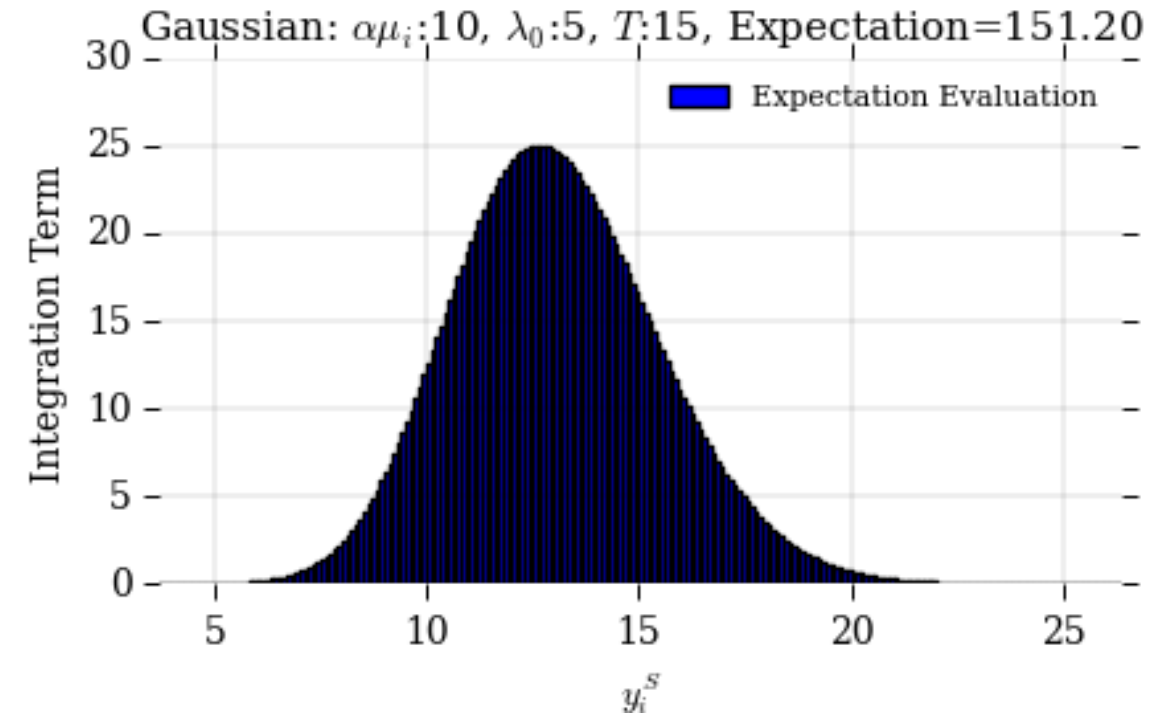

Figure E.8: Moderate noise, partially saturated, low photon count.

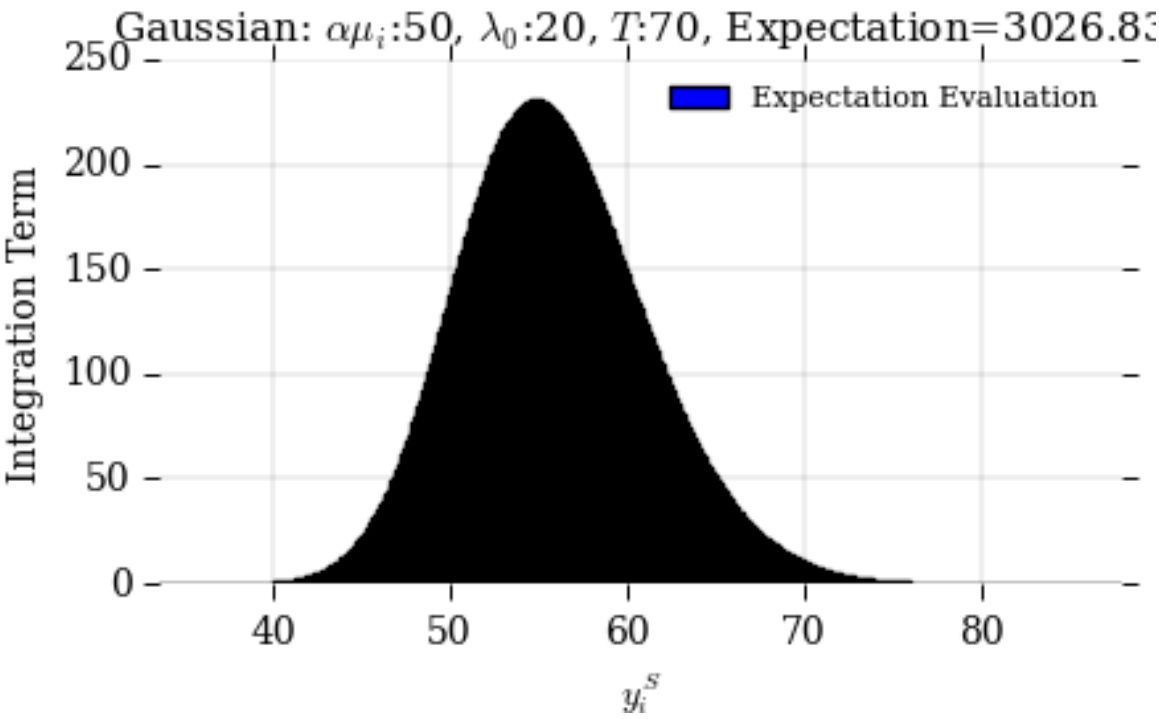

Figure E.9: Moderate noise, partially saturated, high photon count.

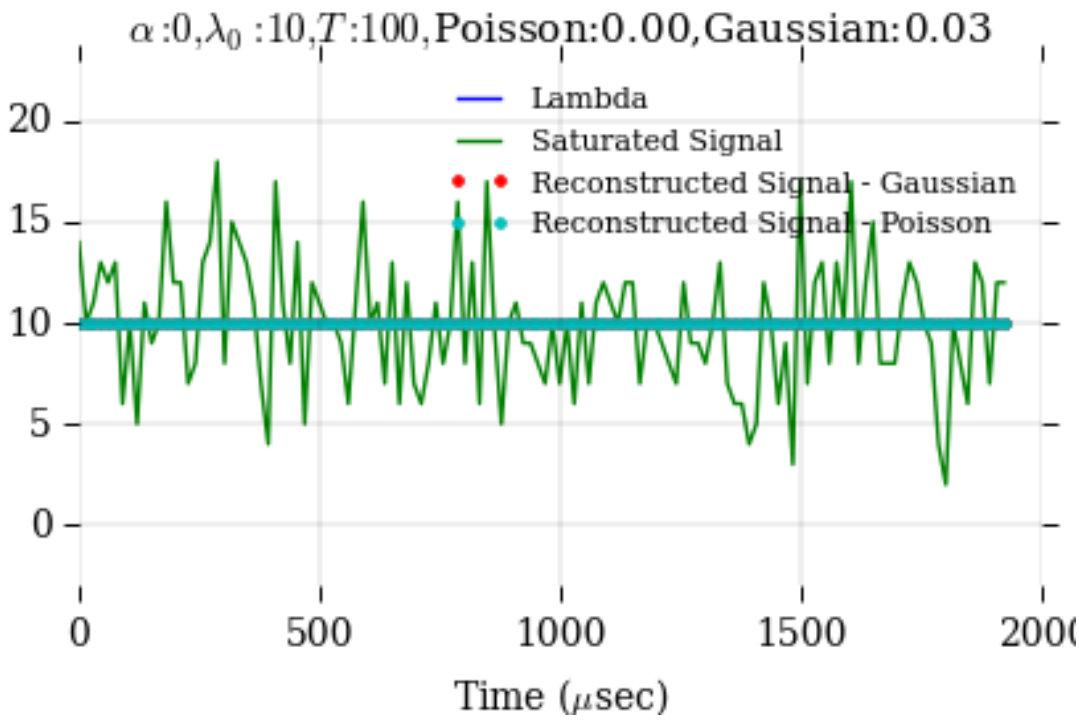

Figure F.1: $\alpha\mu_i = 0$.

APPENDIX F

ALGORITHMIC PERFORMANCE PLOTS

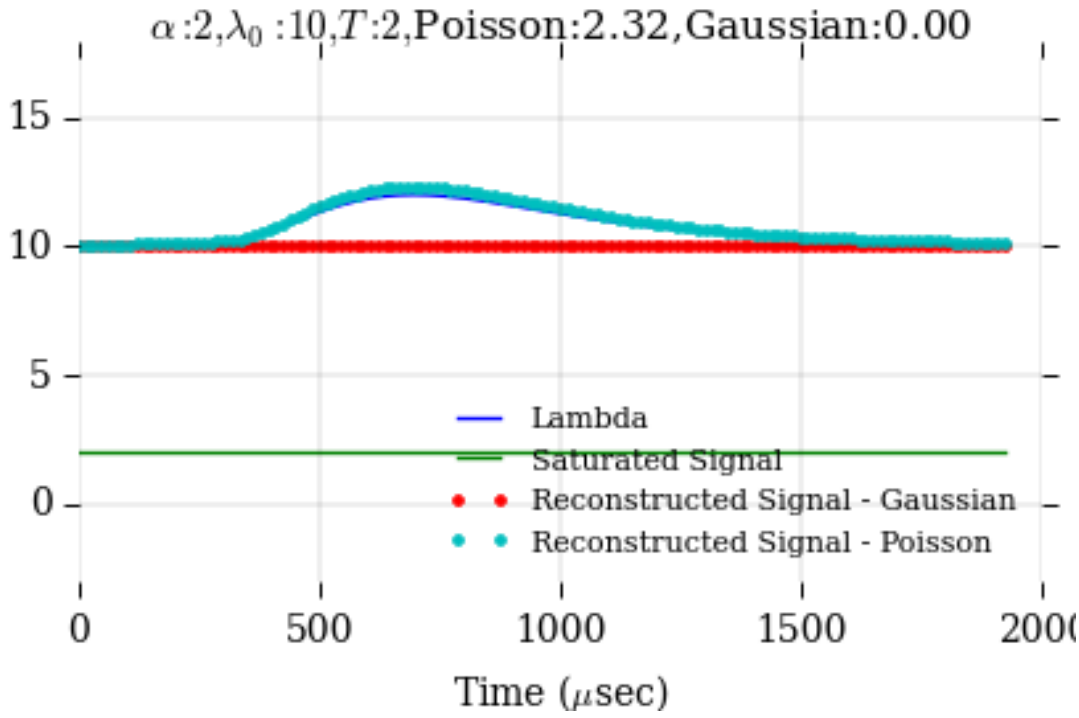

Figure F.2: High noise, entirely saturated, low photon count.

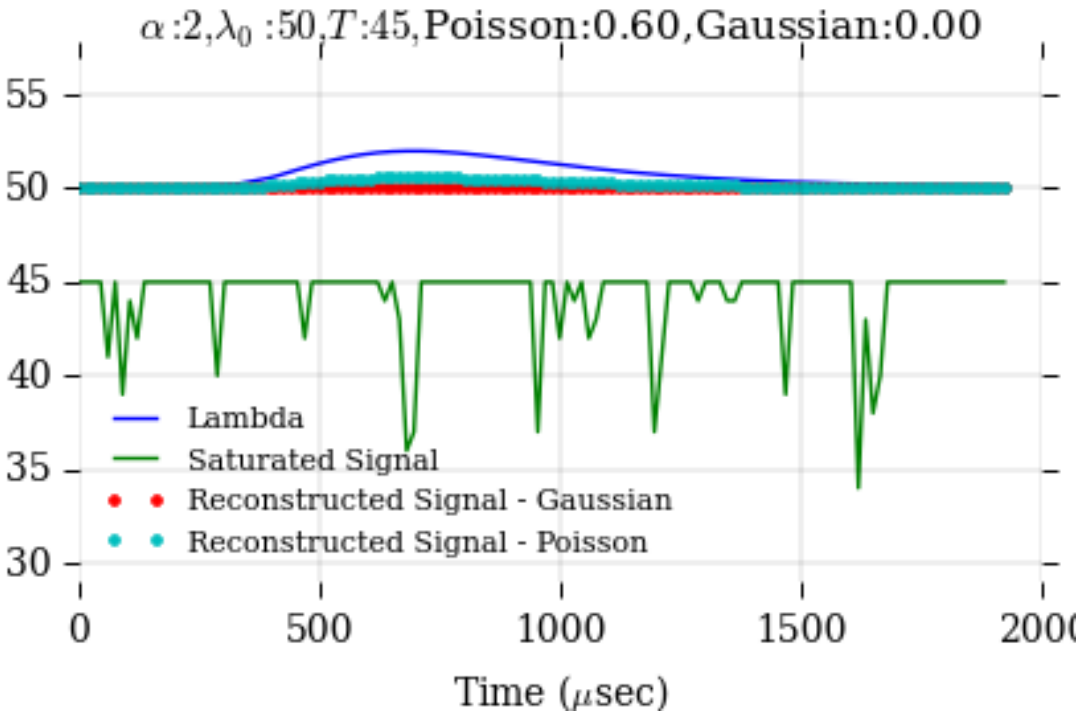

Figure F.3: High noise, partially saturated, high photon count.

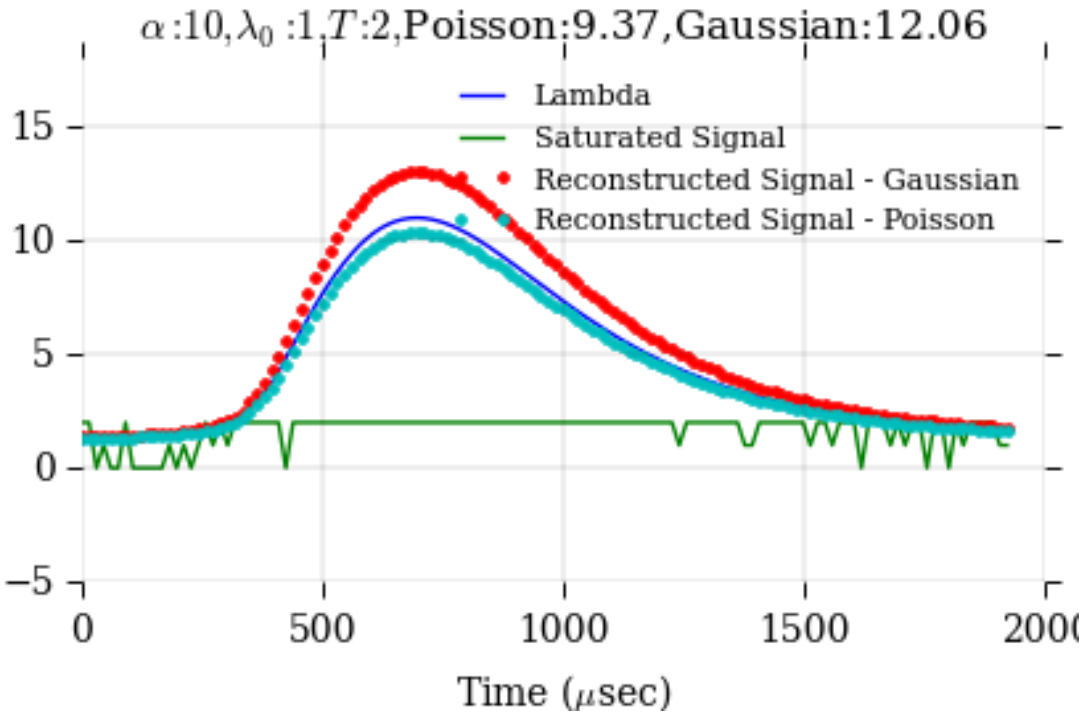

Figure F.4: Low noise, mostly saturated, low photon count.

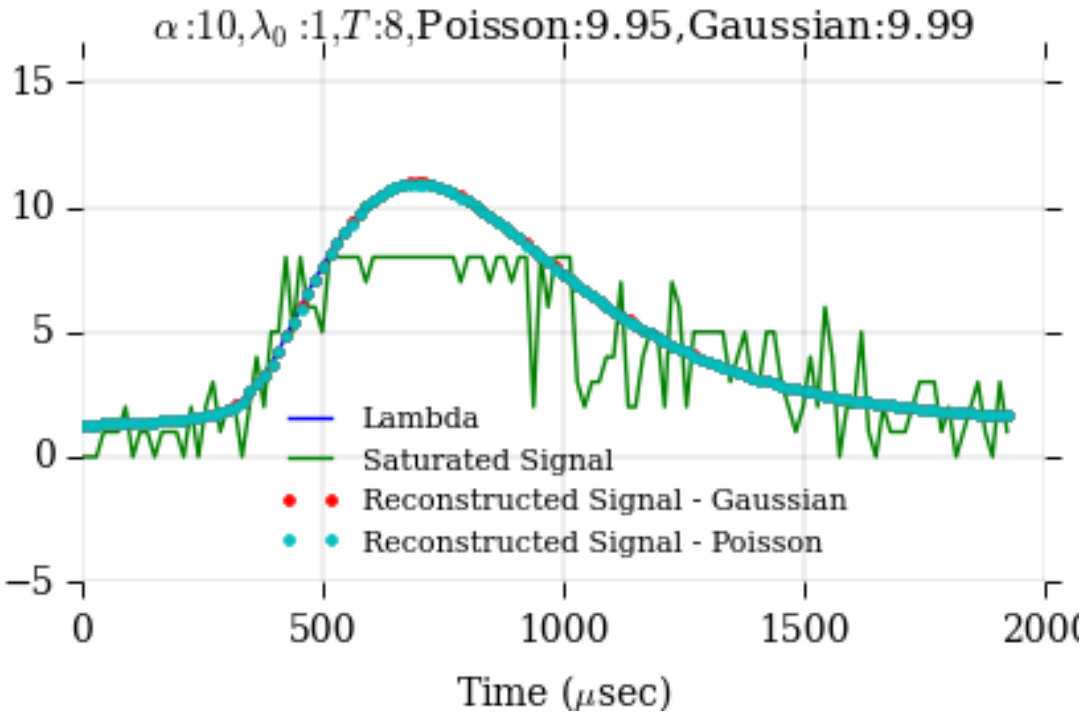

Figure F.5: Low noise, partially saturated, low photon count.

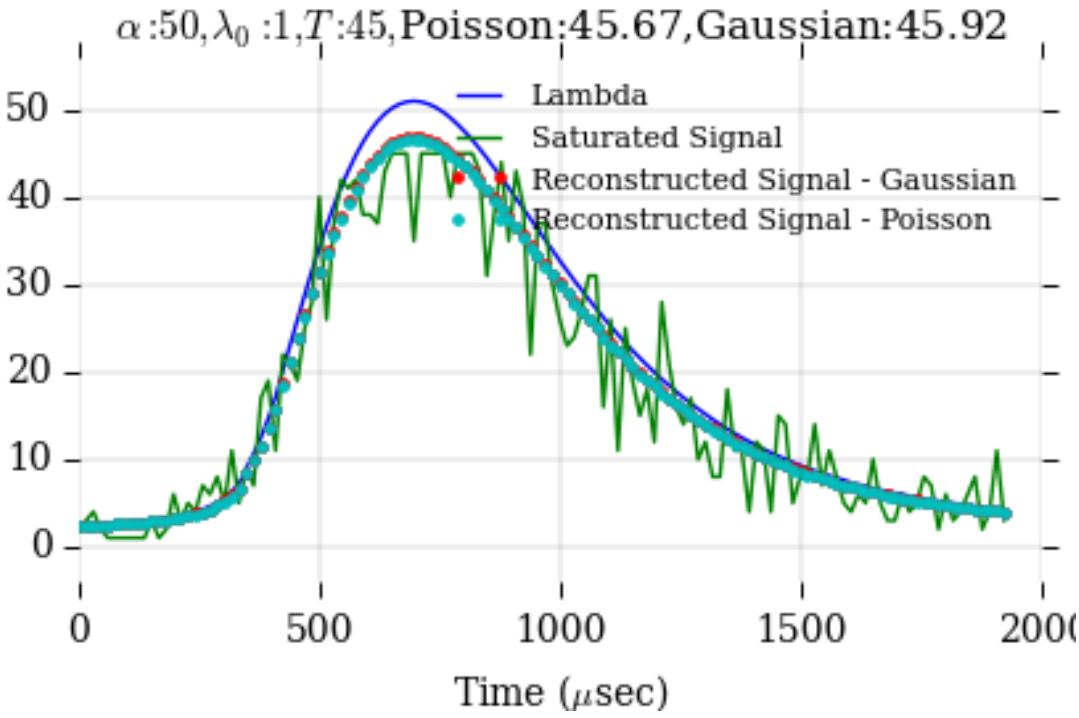

Figure F.6: Low noise, partially saturated, high photon count.

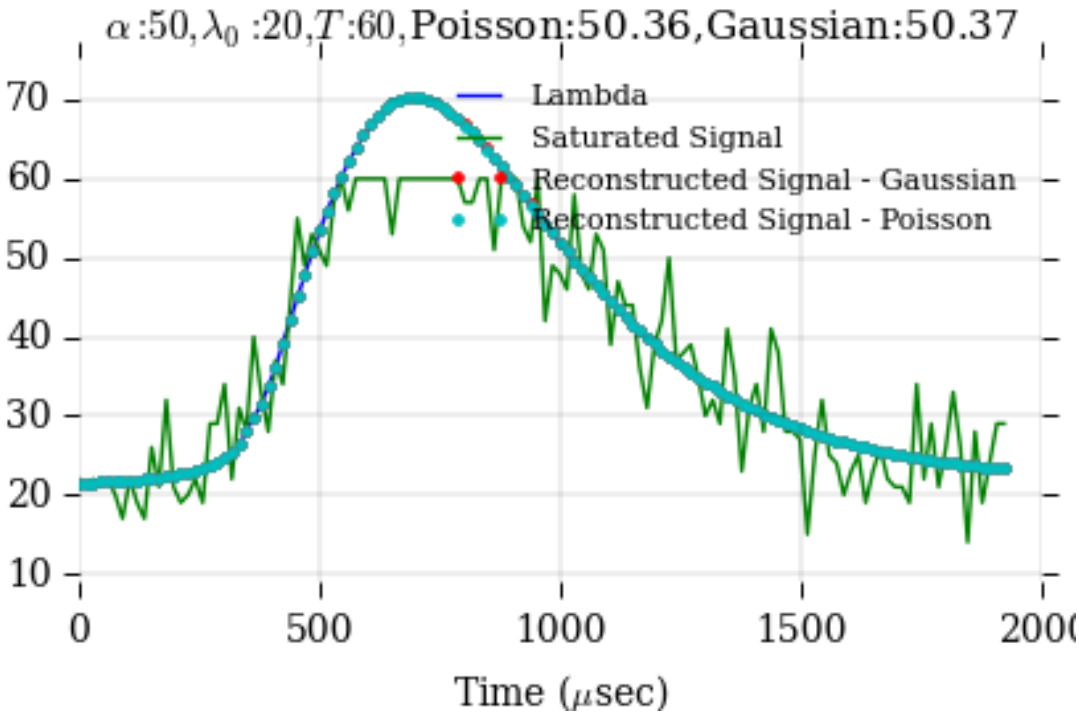

Figure F.7: Moderate noise, partially saturated, high photon count.

REFERENCES

[1] S. M. Kay, *Fundamentals of statistical signal processing : estimation theory*, ser. Prentice Hall Signal Processing Series. Upper Saddle River, New Jersey: Prentice Hall PTR, 1993, vol. 1.

[2] K. L. B. w. Z. T. Harry L. Van Trees, *Detection, Estimation, and Modulation Theory*, second. Hoboken, New Jersey: John Wiley & Sons, Inc., 2013, vol. Part 1 - Detection, Estimation, and Filtering Theory.

[3] G. Seco and J. A. Fernndez-Ruhio, "Maximum likelihood time-of-arrival estimation using antenna arrays: application to global navigation satellite systems," in *Signal Processing Conference (EUSIPCO 1998), 9th European*, 1998, pp. 1–4.

[4] B. I. Erkmen and B. Moision, "Maximum likelihood time-of-arrival estimation of optical pulses via photon-counting photodetectors," in *2009 IEEE International Symposium on Information Theory*, 2009, pp. 1909–1913.

[5] R. M. Marino, J. Richardson, R. Garnier, D. Ireland, L. Bickmeier, C. Siracusa, and P. Quinn, *Photon-counting lidar for aerosol detection and 3d imaging*, 2009.

[6] S. Hernandez-Marin, A. M. Wallace, and G. J. Gibson, "Bayesian analysis of lidar signals with multiple returns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2170–2180, 2007.

[7] N. Antoniadis and A. O. Hero, "Time-delay estimation for filtered poisson processes using an em-type algorithm," *IEEE Transactions on Signal Processing*, vol. 42, no. 8, pp. 2112–2123, 1994.

[8] R. Diamant, R. Kastner, and M. Zorzi, "Detection and time-of-arrival estimation of underwater acoustic signals," in *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2016, pp. 1–5.

[9] H. J. Christian, R. J. Blakeslee, D. J. Boccippio, W. L. Boeck, D. E. Buechler, K. T. Driscoll, S. J. Goodman, J. M. Hall, W. J. Koshak, D. M. Mach, and M. F. Stewart, "Global frequency and distribution of lightning as observed from space by the optical transient detector," *Journal of Geophysical Research: Atmospheres*, vol. 108, no. D1, ACL 4–1–ACL 4–15, 2003, 4005.

[10] C. Buccella, S. Cristina, and A. Orlandi, "Frequency analysis of the induced effects due to the lightning stroke radiated electromagnetic field," *IEEE Transactions on Electromagnetic Compatibility*, vol. 34, no. 3, pp. 338–344, 1992.

[11] H. Wei, Z. Kai-Hong, W. Ling-Hui, and H. Zheng, "Lightning current models' characteristics based on continuous wavelet transform," in *Lightning (APL), 2011 7th Asia-Pacific International Conference on*, 2011, pp. 240–243.

[12] "Forte lightning data," Sandia Natinoal Laboratory, Tech. Rep., ca 2003.

[13] T. E. Light, D. M. Suszcynsky, M. W. Kirkland, and A. R. Jacobson, "Simulations of lightning optical waveforms as seen through clouds by satellites," *Journal of Geophysical Research: Atmospheres*, vol. 106, no. D15, pp. 17 103–17 114, 2001.

[14] D. L. Snyder, R. L. White, and A. M. Hammoud, "Image recovery from data acquired with a charge-coupled-device camera," *J. Opt. Soc. Am. A*, vol. 10, no. 5, pp. 1014–1023, 1993.

[15] D. L. Synder and M. Miller, *Random Point Processes in Time and Space*, second. Springer-Verlag New York, Inc., 1991.

[16] W. B. T.E.L. Light S.M. Davis and A. Jacobson, "Global nighttime lightning flash rates and characteristics observed with the forte satellite," Los Alamos National Laboratory, Sandia National Laboratory, and Niagara University, Tech. Rep., 2003.

[17] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer Science + Business Media, LLC, 2006.

[18] J. Romberg, "Structured marix factorization," Georgia Tech ECE 6254 Spring 2016 class notes, 2016.

[19] D. B. R. A. P. Dempster N. M. Laird, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[20] R. D. Palkki and A. D. Lanterman, "Minimum description length approach to detecting chemicals via their raman spectra," *Optical Engineering*, vol. 50, no. 8, pp. 083 601–083601–10, 2011.

[21] C. Park and S. B. Lee, "Parameter estimation from censored samples using the expectation-maximization algorithm," *Statistics*, 2012.

[22] C. Viwatwongkasem, "{em} algorithm for truncated and censored poisson likelihoods," *Procedia Computer Science*, vol. 86, pp. 240 –243, 2016, 2016 International Electrical Engineering Congress, iEECON2016, 2-4 March 2016, Chiang Mai, Thailand.

[23] D. L. Snyder, C. W. Helstrom, A. D. Lanterman, R. L. White, and M. Faisal, "Compensation for readout noise in ccd images," *J. Opt. Soc. Am. A*, vol. 12, no. 2, pp. 272–283, 1995.

[24] B. L. S. P. Rao, "Remarks on cramer-rao type integral inequalities for randomly censored data," *Lecture Notes-Monograph Series*, vol. 27, pp. 163–175, 1995.

[25] J. Wyckoff and M. Engelhardt, "Cramr-rao lower bounds for estimators based on censored data," *Communications in Statistics - Theory and Methods*, vol. 9, no. 13, pp. 1385–1399, 1980. eprint: http://dx.doi.org/10.1080/03610928008827967.

[26] A. Hero and J. A. Fessler, "A recursive algorithm for computing cramer-rao-type bounds on estimator covariance," *IEEE Transactions on Information Theory*, vol. 40, no. 4, pp. 1205–1210, 1994.

[27] L. Meng, "Method for Computation of the Fisher Information Matrix in the Expectation-Maximization Algorithm," *ArXiv e-prints*, Aug. 2016. arXiv: 1608.01734 [stat.CO].